

Ensembled Self-Adaptive Fuzzy Calibration Models for On-line Cloud Point Prediction

C. Cernuda¹ E. Lughofer¹ P. Hintenaus² W. Märzinger³ T. Reischer⁴ M. Pawlicek⁵ J. Kasberger⁵

¹Department of Knowledge-Based Mathematical Systems, Johannes Kepler University Linz, Austria

²Software Research Center, Paris Lodron University Salzburg, Austria

³i-RED Infrarot Systeme GmbH, Linz, Austria

⁴Dynea Austria GmbH, Krems, Austria

⁵Recendt GmbH, Linz, Austria

Abstract

In this paper we investigate the usage of non-linear chemometric models, which are calibrated based on near infrared (FTNIR) spectra, in order to increase efficiency and to improve quantification quality in melamine resin production. They rely on fuzzy systems model architecture and are able to *incrementally adapt* themselves during the on-line process, resolving dynamic process changes, which may cause severe error drifts of static models. The most informative wavebands in NIR spectra are extracted by a new variant of forward selection, termed as *forward selection with bands (FSB)* and used as inputs for the fuzzy models. A specific *ensemble strategy* is developed which is able to properly compensate noise in repeated spectra measurements. Results on high-dimensional data from four independent types of melamine resin show that 1.) our fuzzy modeling methodology can outperform state-of-the-art chemometric modeling methods in terms of validation error, 2.) the ensemble strategy is able to improve the performance of models without ensembling and 3.) incremental model updates are necessary in order to prevent drifting residuals.

Keywords: cloud point prediction, self-adaptive fuzzy calibration models, ensemble strategy, drift prevention

1. Introduction

1.1. Motivation and State-of-the-Art

In nowadays chemical industrial systems, there is an increasing demand and complexity in monitoring and supervision of processes due to an increasing number of chemical substances and concentrations. Typically, operators are requested to manually draw samples from the process from time to time which are either directly quantified or further analyzed in an off-line stage (e.g. by experts in a laboratory). In many cases, both, the efficiency and quality of chemical systems can be improved by applying online analytic technologies [1], fully automatizing the quantification of the substances in chemical processes [2]. Thereby, the connection of a spectroscopic measurement method [3] in conjunction with chemometric models [4] plays a major role. In the considered process of *melamine resin production*, the essential process parameter to be regularly supervised is the

cloud point. This parameter provides information about the progress of the condensation process in melamine resin production. Monitoring the value of that parameter indicates the best point of time to turn off heating in order to stop the condensation. Currently, the situation is as follows: an operator has to regularly draw samples from the process, whereas for each sample he needs approximately 1-2 minutes in order to obtain the concrete value of the cloud point. Due to the high effort of sample supervision and analysis, a continuous visualization of the condensation process is currently not achieved.

In order to automatize the supervision process and to improve the quantification efficiency and quality of the process parameter, a Fourier-transform-near-infrared (FTNIR) spectrometer was installed at the company for extracting spectra information from the process, reflecting the chemical composition in the melamine resin. The spectra can be used as input data to set up chemometric models for predicting the cloud point. Standard partial least squares method [5] [6] was applied from the PLS toolbox¹ and could achieve a solid (but not ideal) accuracy, which will serve as benchmark for our newly designed chemometric models. Expected implicit non-linearities in the process as well as the fact that the process shows a dynamically changing behavior over time, require the usage of enhanced non-linear chemometric models which are able to adapt their parameters or even structures (if required) on-line. The combination of these aspects prohibits the use of standard chemometric modeling tools.

1.2. Our Approach

In order to resolve aforementioned demands, we present a chemometric modeling approach, which homogeneously joins the following properties in one calibration framework:

- *Non-linear modeling component* which is based on fuzzy systems architecture and which employs statistical information criteria for dimension reduction; therefore, we propose an extended variant of forward selection in order to extract wavebands with arbitrary widths instead of single wavelengths (as done in preliminary chemometric studies, see e.g. [7]), termed as *forward selection with bands*

¹http://www.eigenvector.com/software/pls_toolbox.htm

(FSB). Moreover, we will also connect fuzzy systems with PLS (partial least squares), achieving a sort of a non-linear PLS version.

- Incremental model updates including adaptation of parameters as well as evolution of structural components on demand and on-the-fly; this is necessary in order to react on process drifts over time due to changing dynamics, preventing time-intensive recalibration cycles \rightarrow *self-adaptivity*
- Additionally, we will employ a model *ensembling strategy* for compensating the noise in repeated measurements used to obtain one target sample value.

Due to the combination of these issues, we may also speak about *ensembled self-adaptive calibration models*.

Our approach will be applied to on-line process data from four different types of melamin resins and will show 1.) that our non-linear modeling methodology can outperform state-of-the-art modeling methods in terms of validation error, 2.) that the usage of ensemble strategy is able to improve the performance of models without ensembling significantly and 3.) incremental model updates are necessary in order to keep the predictive quality of the models high during the further ongoing on-line process.

2. Chemometric Modeling Steps

2.1. Non-Linear Fuzzy Modeling and Waveband Extraction

2.1.1. Architecture and Training (Outline)

For modeling implicit non-linearities contained in the melamine resin production process, we exploit the Takagi-Sugeno (TS) fuzzy model architecture [8], which have some favorable properties [9]:

- Universal approximation: ability to resolve any degree of non-linearity with sufficient accuracy.
- Employing piece-wise linear predictors in the form of hyper-planes $l_i = w_{i0} + w_{i1}x_1 + w_{i2}x_2 + \dots + w_{ip}x_p$ $i = 1, \dots, C$ (assuming p features) for modeling the behavior in C partial local regions: this achieves some sort of synergies to local weighted regression models or multi-model partial least squares, especially when connected with PLS.
- Combining the piecewise local linear predictor with normalized multivariate Gaussian kernels to form an over-all smooth and differentiable global model:

$$\hat{f}(\vec{x}) = \hat{y} = \sum_{i=1}^C l_i \Psi_i(\vec{x}) \quad \Psi_i(\vec{x}) = \frac{\mu_i(\vec{x})}{\sum_{j=1}^C \mu_j(\vec{x})} \quad (1)$$

with $\mu_i(\vec{x})$ the membership degree of x to the i th kernel.

Non-linearity plays a more and more major role in today's chemical processes [4], linear predictors are often beneficial in cases where (off-line) calibration samples

are hard to obtain and smooth functions have some favorable analytical properties. Takagi-Sugeno fuzzy systems offer an architecture, which is able to flexibly handle the non-linearity requested and the non-linearity afforded; this is controlled by the index C in the sum of (1), i.e. the number of model components used. In case of $C = 1$, the Takagi-Sugeno fuzzy model in (1) automatically reduces to a multivariate linear regression model. Thus, using latent variables as inputs, it can also turn to a global PLS model.

2.1.2. New Waveband Selection for Noise Reduction

One novelty in this paper refers to the dimensionality reduction process, which is essential for models with implicit local structures, as suffering significantly from curse of dimensionality effect [9]. Especially, in the case of NIR spectra, usually the dimensionality (=number of wavelengths) lies in the range of 1000-3000, depending on the spectrometer used [3] (in our application it will be 1249), which is un-managable high for fuzzy systems architecture. In [7], we reported a wavelength selection process, which is based on a variable ranking scheme exploiting some sort of *forward selection with orthogonalization* [10]. However, it turned out for some applications (and also for that one treated in this paper) that this approach is prone to noise levels occurring during measurement recordings. The basic drawback comes in due to selecting single and disjoint wavelengths (positions) instead of larger wavebands, containing consecutive wavelengths. Therefore, we design a new selection scheme, termed as *FSB = Forward Selection with Bands*, where in each iteration still the wavelength correlated most to the target y is selected, i.e.:

$$j^* = \operatorname{argmax}_{j=1, \dots, n_{\text{wave}}} (R_{j,1}^2 = R^2(\text{wave}_j, y)) \quad (2)$$

with R^2 the classical R -squared statistics, however adjacent wavelengths are adjoined until a certain saturation of regression model quality is reached:

$$R_{j,i+1}^2 - R_{j,i}^2 \leq R_{j,1}^2 \quad (3)$$

where the first index denotes the band number and the second one denotes the width. We constrain this iterative joining by a maximal number of wavelengths in one waveband max_wave in order to prohibit large span bands. Figure 1 visually shows (from (a) to (d)) how single wavelengths are step-wise included to form a waveband. Once a band waveband_i is found, its contribution for explaining the (remaining) target y_{rem} is subtracted: a regression model \hat{f} is built using all wavelengths in all so-far selected bands $\text{waveband}_{1, \dots, i}$ and the following calculated:

$$\vec{y}_{\text{rem}} = \vec{y} - \hat{f}(X_{\text{red}}) \quad (4)$$

This is repeated until max_dim wavebands (containing a different number of wavelengths) are selected and a ranking of importance across wavebands is achieved (first selected = most important one and so on). In a first trial-and-error phase, this scheme turned out to perform not really well. The problem was that wavebands may

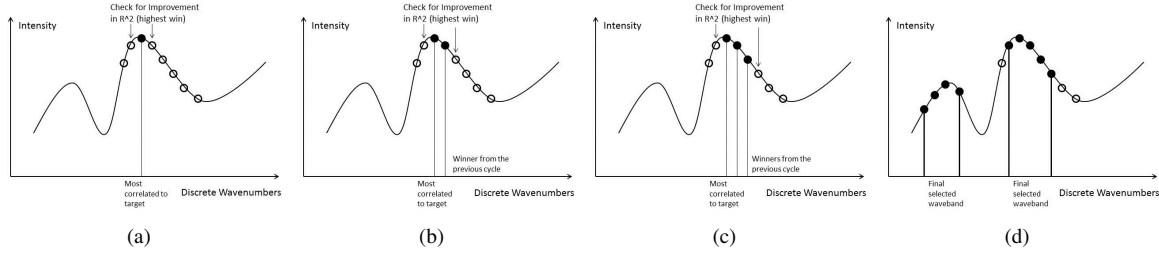


Figure 1: Waveband construction according to our new selection approach termed as *Forward Selection with Bands (FSB)*

contain up to n wavelengths, i.e. whenever max_dim (usually set to 10-20) wavebands are selected, a high number of wavelengths was included in the modeling process. However, fuzzy systems treat each wavelength in a waveband as separate dimension in the rule product space, such that the curse of dimensionality effect is still severe. Therefore, we investigated a post-processing phase where we projected each waveband to a single latent variable with the help of robust partial least squares [6], obtaining a ranked list ($lat_1, \dots, lat_{max_dim}$) of inputs for the fuzzy model.

The fuzzy models training procedure (FLEXFIS) will be also connected with standard PLS, which is conducted prior to the training process in order to extract latent variables on a global basis

2.2. Ensemble Technique for Boosting Performance

In literature, ensembling techniques enjoy a wide attraction in order to boost performance of classification and regression models [11]. Basically, there are two major research lines in the field of ensemble methods: 1.) training of so-called *weak models*, usually on subset of features or on parts of the target space, and combining these weak models to form a strong over-all model with high performance [12]; or 2.) using a different set of (full) regression model architectures, exploiting their diversity which plays a central role in order to guarantee an increase of the stand-alone performance of single models [13].

In our application, we are confronted with a different ensembling problem, which stems from the fact that for each process state three repeated measures are recorded by the data acquisition machine, which all show the same target variable values, but slightly different spectra according to noise variations in the recordings. Thus, the goal of our ensembling method is to perform a kind of noise reduction. We therefore build regression models for each of the recordings separately, i.e. using all samples from the each of the repetitions $\rightarrow S_{1,2,3}$, and train a model for each set separately. Noise reduction is then achieved by a weighted combination of the single model outputs, rather than integrated in the learning process of one global model. Thus, our noise reduction approach by ensembling can be also seen as a variance reduction of one single model. In particular, we define

$\forall i = 1, 2, 3:$

$$S_i = \{\vec{x}_j | \vec{x}_j \in X \wedge j \in \{1, \dots, N\} \wedge \text{mod}(j, 3) = 3 - i\} \quad (5)$$

with X the whole calibration matrix containing the samples in the same order as they were recorded and mod the modulo function. Then, we define the regression models by $\mathbb{R}_i \leftarrow \mathbb{T}(S_i)$ with \mathbb{T} a training procedure operating on set S_i . The final predicted model output is achieved by a weighted combination in form of:

$$\hat{y} = w_1 * \mathbb{R}_1 + w_2 * \mathbb{R}_2 + w_3 * \mathbb{R}_3 \quad (6)$$

We used a general approach for eliciting the weights, namely by calculating model qualities pointing to their predictive generalization power. Thus, within a K -fold cross-validation procedure (we set $K = 10$ in all our experiments), we elicited the 'cross-validated' R^2 value as:

$$R_i^2 = \frac{1}{K} \sum_{j=1}^K R^2(j) \quad \forall i = 1, 2, 3 \quad (7)$$

with $R^2(j)$ the R -squared value for the j th fold. The R^2 values calculated in this way are reliably reflecting the general quality of the model: this is in accordance to the fact that a CV estimator is a good indicator for the generalization quality of a model, see for instance [14] (Chapter 7). Then, the weights in (6) are elicited by:

$$w_i = \frac{R_i^2}{R_1^2 + R_2^2 + R_3^2} \quad \forall i = 1, 2, 3 \quad (8)$$

ensuring that the degree of contribution of the prediction of each model corresponds to the portion of its generalized (cross-validated) model quality.

2.3. Reacting on Time-Dependent Process Drifts

A specific problem in melamine resin production originates from process drift which may make chemometric models initially built from pre-recorded calibration samples (representing a particular composition, environmental actuality etc.) more and more "out-dated" over time, thus their predictions are becoming more and more uncertain. A typical example of drifts from the real melamine production process is shown in Figure 2, visualizing residuals (=deviations between predicted and observed values of cloud point y) within a 3 weeks period,

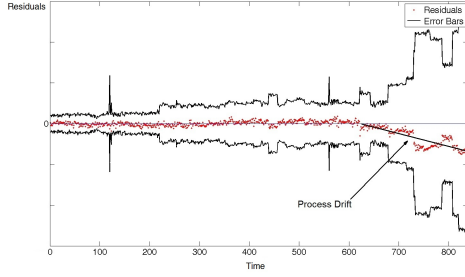


Figure 2: Examples of long term process drifts at melamine resin production, please note the increasing residuals (dots) at the end of the streams and the synchronously widening of prediction uncertainty modeled by local error bars

starting right after the generation of the initial chemometric models.

Although the process drifts are usually in the long-term run (starting after around 2 weeks), a time-intensive manual re-calibration of the models is often not a reliable option. Thus, we propose an automatic dynamic model update scheme, which adapts model parameters and expands as well as shrinks model structures on-the-fly. This will be achieved in an incremental single-pass learning context, following the natural behavior of on-line data streams [15]. Single-pass learning [16] guarantees a fast model update and minimal virtual memory usage as no prior data is used; the memory usage can be essential in case of on-line instruments/spectro-meters integrating micro-controllers with reduced RAM.

In the following two sub-sections, we will summarize the update mechanisms for the TS fuzzy systems as defined in 2.1.1. Additionally, we will give an out-line of an incremental PCR (*principal component regression*) approach (denoted as *incrPCR*), representing a recursive global linear method without evolving structural components (just parameter updates).

2.3.1. Evolving and Shrinking TS Fuzzy Systems

The update of TS fuzzy systems is conducted with the usage of the FLEXFIS++ learning engine [17], which, as a regression method, is part of the *FLEXFIS* family and operates on a single-pass incremental manner, including structural evolution and recursive parameter update, as is a common denominator in the context of *evolving fuzzy systems (EFS)* [18]. Combined with spectral samples, we may speak about *evolving (non-linear) chemometric models*.

A basic difference to the original approach is that, when a new spectral sample \vec{x} comes in, first it is projected on the subspace generated by p latent variables. In the case of *FLEXFIS+FSB* variant, they correspond to the p most important wavebands found with the algorithm described in Section 2.1.2:

$$\vec{x}_{proj} = \vec{x} * [\vec{w}_{B_1,1} | \dots | \vec{w}_{B_p,1}] \quad (9)$$

In *FLEXFIS+PLS* variant, the p latent variables are obtained applying PLS in a global manner:

$$Lat_p = [\vec{lat}_1 | \dots | \vec{lat}_p] = X * W_{X,p} \Rightarrow \vec{x}_{proj} = \vec{x} * W_{X,p} \quad (10)$$

The new projected sample \vec{x}_{proj} is then sent into the update mechanism of the learning engine, which is decomposed into the following parts:

1. Check if the current sample fits into the cluster partition, where one cluster corresponds to one rule (out of C rules in (1)).
2. If yes: the current partition is updated in a convergent manner, i.e. with a decreasing learning gain over life-time and support of the cluster. This assures *stability*.
3. If no: a new rule = cluster is added to the rule base as the new sample falls into an unexplored region of the feature space. This assures *plasticity*.
4. Adding corrections terms to consequent parameters and inverse (parameter) covariance matrices to balance out non-optimal situations in the RWLS estimator.
5. Update the consequent functions and inverse (parameter) covariance matrices with the usage of RWLS [19], thus achieving a local learning scheme (for each rule separately) [20].

A specific property of our approach (compared to other EFS approaches [18]), is the fact that it guarantees sub-optimality which is close to optimality in terms of the least squares error functional (Step 4 above).

In order to compensate highly dynamic changes in the on-line stream, leading to extraordinary rule movements, we investigate additional concepts:

- Merging of strongly over-lapping rules: a fast extended kernel-based similarity measure is used to estimate the degree of overlap in $[0, 1]$:

$$S_{ker}(A, B) = (e^{-\frac{(c_A - c_B)^2}{\sigma_A^2 + \sigma_B^2}} + e^{-(c_A - c_B)^2 - (\sigma_A - \sigma_B)^2}) / 2 \quad (11)$$

with c_A the center of fuzzy set A and σ_A is characteristic spread. The first term is motivated from statistical theory, comparing whether two Gaussian distributions are different or not (hypothesis test) [21]. The second term assures that smaller sets within larger sets are not merged (as representing different specifications of data clouds in different parts of the high-dimensional feature space). If the fuzzy sets of all antecedent parts of two rules are similar (to a default degree of at least 0.8), it points to redundant local partitioning and the rules can be merged.

- Forgetting of older learned relations over time in order to be able to bring in more flexibility to adjust to new changed situations, especially in case of drift occurrences. We integrate *forgetting factors* 1.) in the consequent learning achieving a smooth exponential out-weighting over time and

2.) in the antecedent learning part by reactivating converged rules. The methodology is leaned on the approaches used in [22].

- Elimination of obsolete rules: in more extreme drift cases, also referred as *data shifts*, a new rule is already opened up within the shifted data cloud. However, older rules may still contribute (at least a little) to the final model output, especially when using multi-dimensional kernels with infinite support: in fact all rules fire to a certain degree in (1). *Rule ages* [23] may be a plausible concept for tracking out-dated rules which are no longer attached. However, this rely on unsupervised criteria and thus do not take into account the impact of the rules onto the final model output. In this paper, we propose the concept based on an influence of a fuzzy rule in the data stream learning context over time, compared to the influence of all rules, employing the normalized membership degrees and the consequent hyper-planes [24]. Whenever the influence drops below a certain threshold, it can be seen as relatively insignificant compared to the other rules.

2.3.2. Incremental PCR

Alternatively to the evolving TS fuzzy systems, we propose an incremental update mechanism for PCR (incPCR) [25], as serving as one of the most widely used and renowned state-of-the-art modeling technique in the field of chemometrics. The update engine in incPCR is driven by two aspects:

- Updating the eigen-space (eigenvectors and eigenvalues) in order to permanently adjust the rotated axes to be ideal in terms of pointing into those directions where maximal variance of sample data appears.
- Updating the regression coefficients $\vec{\beta}$ obtained in batch mode by $\vec{\beta} = (S_m^T S_m)^{-1} S_m^T y$, with $S_m = P_m X$ the scores by projecting each original sample from X to all selected components in $P_m = [p_1, \dots, p_m]$. in order to account for changes in the dependencies between principal components and the target variable; in practice, this means to move, rotate the regression hyper-plane spanned by $[\beta_1, \dots, \beta_m, y]$.

A key point for a well-posed update of the eigen-space there is to check whether a new incoming sample, x_{new} , has almost all energy (90%) in the current eigen-space $P_m = [p_1 p_2 \dots p_m]$ or not. This can be done by computing the residual vector \vec{h} with [26]

$$\vec{h} = (x_{new} - \bar{x}) - P_m \vec{g} \quad (12)$$

where \bar{x} denotes the vector obtained by averaging the input samples and \vec{g} is calculated by $\vec{g} = P_m^T (x_{new} - \bar{x})$, and then checking whether the norm of the residual vector is smaller than a threshold value η . If this is the case, the eigen-space is simply updated, otherwise a column is added to P_m and $m = m + 1$, see [26]. The regression coefficients are updated with the usage of recursive least squares (RLS) approach [19], afterwards, in order to account for changing optimal positions of hyper-planes.

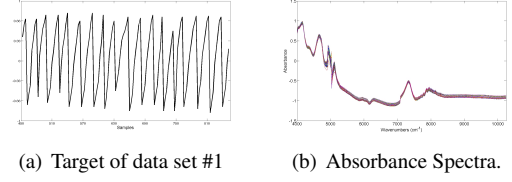


Figure 3: Visualization of data set #1.

3. Experimental Setup

3.1. Data set description

The data sets provided by our partner company (Dynea) were recorded with the usage of FTNIR process spectrometer which has been optimized from the beginning of the development with respect to robustness, measurement time and cost-effectiveness and is designed on bases of a Michelson interferometer. In order to obtain a robust design, a monolithic interferometer [27] is utilized in the Michelson setup. The measurement is conducted by a transmission probe (Hellma) with an optical path of 1 mm through the fluid. The final drawn data sets consist of multiple instances, i.e. every instance is a set of 3 consecutive measurements. The number of wavelengths (variables) and instances vary from one formula to the other. Table 1 summarizes the characteristics of the data sets for the four different recipes under consideration; these are drawn from different independent production cycles, thus offering significant diversity for validating our approach.

Figure 3 shows respectively the measured cloud point values for the multiple instances data set #1, 3(a), and the spectral values as a representation of the multiple instances, in 3(b). Furthermore, process variables were recorded by sensor readings (temperature, pressure, pH value) and synchronized with the recordings of the NIR spectrometer recordings. The original goal was to provide orthogonal or at least independent information to NIR; however, it turned out that they were more or less perfectly correlated (> 0.99 correlation coefficient) with some bands in the spectra.

3.2. Evaluation Scheme

3.2.1. Batch Phase

The first stage evaluates our new proposed methods (FLEXFIS+FSB/PCA/PLS with and without ensembling) within a batch off-line process, using RMSE 10-fold cross-validation error as comparison measures

$$CV_RMSE = \sqrt{\frac{1}{K} \sum_{i=1}^K \frac{\sum_{k=(i-1)*(N/K)+1}^{i*(N/K+1)} (y(k) - \hat{y}(k))^2}{N/K}} \quad (13)$$

with N the number of calibration samples, and the standard deviation of the RMSE over different folds:

$$SD = \sqrt{\frac{1}{K-1} \sum_{i=1}^K (RMSE(i) - CV_RMSE)^2} \quad (14)$$

Recipe	#1	#2	#3	#4
Instances Training	1608 (3 x 536)	1269 (3 x 423)	753 (3 x 251)	1119 (3 x 373)
Wavelengths	1249	1249	1249	1249

Table 1: Data sets summary.

with $RMSE(i)$ the root mean squared error over the i th fold. The latter provides a measure how sensitive a method reacts onto a specific fold combination. For benchmark purposes, we apply widely used and renowned state-of-the-art approaches including multiple linear regression (MLR), principal component regression (PCR) [25], partial least squares regression (PLSR) [5], local weighted regression (LWR) [28], regression trees (RegTree) [14], stepwise regression (StepwiseReg), ridge regression and GLMnet [29] (a convex combination of Lasso and ridge regression) and FLEXFIS combined with old forward selection (FLEXFIS+FS) variant [7].

Furthermore, all methods are evaluated within a parameter grid scenario, where the most sensitive parameters of these methods (e.g. the pruning level in case of RegTree or the α -value for the sensitivity to add new variables in case of StepwiseReg) are varied according to a pre-defined grid and for each parameter combination a cross-validation procedure (with the same splits) conducted. Finally, for each method that combination is selected which minimizes the following criterion:

$$RMSE_{\alpha}^{(pen)} = RMSE \cdot e^{\alpha \cdot param1 + \beta \cdot (1 - param2)} \quad (15)$$

If an algorithm does not involve $param1$, then it receives the default value $param1 = 0$. Analogously for $param2$ with default value $param2 = 1$. Empirically, we have determined $\alpha = 0.05$ and $\beta = 0.5$ as adequate normalization values. This criterion assures to punish more complex models having a similar RMSE value (error) over others in order to decrease over-fitting effects as much as possible.

3.2.2. Adaptive Phase

The second stage includes the verification of drifts issues and the demonstration of the importance to resolve these issues with incremental adaptive learning methods. For doing so, we compare the quality of static models, initially trained by applying the same procedure as outlined in the previous section (grid search + 10-fold CV), with the quality of dynamic models, permanently updated on new on-line data samples. We take a small initial batch for training and the remaining instances as longer stream for testing, as usually drifts only arise over a longer time period of about at least 2 weeks; three different split portions are used: 10/90, 25/75 and 33/67. In this sense, the three different testing portions also serve as separate validation data for the static models. We will see that static models significantly suffer more and more, the smaller the training portion is. The quality of the models is measured in terms of accumulated RMSE over time; in case of dynamic models, always

	Recipe #1	Recipe #2	Recipe #3
MLR	0.096 ± 0.02	0.566 ± 0.48	0.160 ± 0.02
PCR	0.041 ± 0.03	0.083 ± 0.06	0.029 ± 0.01
PLSR	0.036 ± 0.02	0.044 ± 0.03	0.028 ± 0.01
LWR	0.149 ± 0.04	0.220 ± 0.09	0.190 ± 0.08
RegTree	0.120 ± 0.07	0.125 ± 0.04	0.120 ± 0.07
StepReg	0.039 ± 0.01	0.059 ± 0.06	0.034 ± 0.01
GLMnet	0.033 ± 0.01	0.064 ± 0.07	0.031 ± 0.01
RidgeReg	0.096 ± 0.02	0.714 ± 1.13	0.205 ± 0.06
Fuz.+PCA	0.033 ± 0.01	0.055 ± 0.03	0.032 ± 0.02
Fuz.+PLS	0.029 ± 0.01	0.037 ± 0.03	0.020 ± 0.01
Fuz.+FS	0.137 ± 0.02	0.073 ± 0.04	0.079 ± 0.01
Fuz.+FSB	0.054 ± 0.02	0.063 ± 0.05	0.048 ± 0.01

Table 2: CV results (RMSE+SD, divided by the range of the target) for three recipes

a one-step-ahead error is calculated and accumulated within an incremental model update scheme. The applied methods will be incrPCR (as described in Section 2.3.2), FLEXFIS+PLS and FLEXFIS+FSB incremental (as described in Section 2.3.1) and their static counterparts. Thus, by comparing incrPCR and FLEXFIS, we also verify the impact/necessity of structural evolutions and non-linearities on the model accuracy.

4. Results

4.1. Batch Phase

The results in Table 2 show the CV accuracies and sensitivities of the various methods on the four recipes of Melaminharz, indicating the preferred methods in bold font for each recipe.

A viewpoint on the results in Table 2 is that in case of Recipe #1 FLEXFIS+PLS and GLMnet perform best and similar, whereas for the other three recipes FLEXFIS+PLS can outperform all other state-of-the-art methods. The interpretation of this result is that introducing non-linearity in conventional PLS helps to improve its performance, also boosting it over other state-of-the-art methods. Selecting bands explicitly and projecting each one locally to one latent variable (as done in FLEXFIS+FSB), does indeed not help to improve the error rates further, but is able to outperform the old version FLEXFIS+FS (extracting single wavelengths) on all three data sets.

Table 3 presents the results obtained when employing the ensembling scheme according to Section 2.2, extracting the weights from a cross-validated model quality R^2 . We have also compared the performance of the ensembling strategy with a typical approach when repeated measures are available, that is averaging the spectral information before the modeling process. The best method for each formulation is highlighted in bold font and always corresponds to an ensembling variant, also in most other cases the ensembling brings an improvement over the original non-ensembling, i.e. using all measurements as one data set modeling scheme. It is interesting

Method	Recipe #1	Recipe #2	Recipe #3	Recipe #4
PLSR (orig)	0.0359 ± 0.0195	0.0439 ± 0.0330	0.0284 ± 0.0090	0.0399 ± 0.0359
PLSR (aver)	0.0358 ± 0.0206	0.0403 ± 0.0284	0.0270 ± 0.0089	0.0397 ± 0.0363
PLSR (ens)	0.0340 ± 0.0263	0.0426 ± 0.0335	0.0264 ± 0.0092	0.0379 ± 0.0355
GLMnet (orig)	0.0326 ± 0.0107	0.0638 ± 0.0692	0.0309 ± 0.0069	0.0476 ± 0.0330
GLMnet (aver)	0.0326 ± 0.0106	0.0679 ± 0.0512	0.0324 ± 0.0084	0.0474 ± 0.0382
GLMnet (ens)	0.0321 ± 0.0110	0.0621 ± 0.0927	0.0310 ± 0.0092	0.0437 ± 0.0349
FLEXFIS+PLS (orig)	0.0297 ± 0.0118	0.0370 ± 0.0277	0.0203 ± 0.0065	0.0428 ± 0.0317
FLEXFIS+PLS (aver)	0.0268 ± 0.0117	0.0343 ± 0.0274	0.0166 ± 0.0047	0.0411 ± 0.0334
FLEXFIS+PLS (ens)	0.0242 ± 0.0083	0.0291 ± 0.0212	0.0104 ± 0.0089	0.0281 ± 0.0250
FLEXFIS+FS (orig)	0.1371 ± 0.0179	0.0733 ± 0.0356	0.0786 ± 0.0110	0.0970 ± 0.0414
FLEXFIS+FS (aver)	0.0573 ± 0.0167	0.0620 ± 0.0215	0.0638 ± 0.0143	0.0787 ± 0.0312
FLEXFIS+FS (ens)	0.0482 ± 0.0130	0.0550 ± 0.0215	0.0458 ± 0.0091	0.0695 ± 0.0317
FLEXFIS+FSB (orig)	0.0540 ± 0.0146	0.0626 ± 0.0456	0.0477 ± 0.0082	0.0951 ± 0.0218
FLEXFIS+FSB (aver)	0.0413 ± 0.0171	0.0610 ± 0.0268	0.0464 ± 0.0089	0.0911 ± 0.0240
FLEXFIS+FSB (ens)	0.0395 ± 0.0111	0.0583 ± 0.0398	0.0434 ± 0.0082	0.0601 ± 0.0387

Table 3: CV ensemble results for the four formulations '(ens)' compared with original non-ensembled results '(orig)' as well as with the averaged spectral information of the repeated measurements '(aver)'

Method	10/90	25/75	33/67
Recipe #1			
PCR (stat)	0.0809	0.0807	0.0664
incrPCR (dyn)	0.0555	0.0513	0.0483
FLEXFIS+PLS (stat)	0.0902	0.0780	0.0590
FLEXFIS+PLS (dyn)	0.0703	0.0538	0.0400
FLEXFIS+FSB (stat)	0.1271	0.1155	0.0997
FLEXFIS+FSB (dyn)	0.0551	0.0510	0.0411
Recipe #2			
PCR (stat)	0.8516	0.0977	0.1090
incrPCR (dyn)	0.0682	0.0570	0.0654
FLEXFIS+PLS (stat)	0.7338	0.1008	0.1218
FLEXFIS+PLS (dyn)	0.2113	0.0575	0.0658
FLEXFIS+FSB (stat)	0.1497	0.0969	0.1236
FLEXFIS+FSB (dyn)	0.0638	0.0572	0.0564
Recipe #3			
PCR (stat)	0.4069	0.1731	0.0893
incrPCR (dyn)	0.0605	0.0604	0.0410
FLEXFIS+PLS (stat)	0.2691	0.1674	0.0695
FLEXFIS+PLS (dyn)	0.0654	0.0342	0.0233
FLEXFIS+FSB (stat)	0.1520	0.1491	0.0981
FLEXFIS+FSB (dyn)	0.0745	0.0660	0.0499
Recipe #4			
PCR (stat)	0.2360	0.1629	0.1358
incrPCR (dyn)	0.0733	0.0480	0.0469
FLEXFIS+PLS (stat)	0.2146	0.1602	0.1473
FLEXFIS+PLS (dyn)	0.0572	0.0514	0.0499
FLEXFIS+FSB (stat)	0.2203	0.1045	0.0793
FLEXFIS+FSB (dyn)	0.0711	0.0679	0.0623

Table 4: RMSE values for predicting new on-line samples over time with static models (denoted by 'stat') and incrementally adapted ones as described in Section 2.3 (denoted as 'dyn')

to see that the ensembling method can even improve the quality of the best original models for each formulation: the range of the improvement lies from 19% (in case of #1) up to astonishing 49% (in case of #3). Moreover, averaging the spectral information prior to the modeling process behaves worse than the ensembling procedure (with the exception of PLSR in #2) and a bit better than using the original data set.

4.2. Adaptive Phase

According to the off-line results demonstrated in the preliminary section, for the on-line phase we focussed on adaptive modeling on the original spectra information and using ensemble strategy. The results on the three different training/testing portions are shown in Table 4, which compare static models built up once on the training portion and kept fixed during the whole on-line validation phase with dynamically adaptive models, according to the methodologies described in Section 2.3. Obviously, static models are not able to predict new samples with sufficient accuracy, especially for Recipes #2 to #4, as producing significantly large errors (errors above 10%). Moreover, the errors become

larger, the larger the prediction horizon becomes (tables to be read from last to first column). On the other hand, dynamic models produce much lower errors, partially coming close to the more optimistic off-line CV results, and tendentially remaining quite stable over different time horizons: in bold font, we highlighted the best dynamic method to be used for each data set. For Recipe #1, incrPCR and FLEXFIS+FSB perform equally well, also in the other cases incrPCR performs in a similar stable manner as FLEXFIS+PLS and FLEXFIS+FSB, with little worse accuracy as is also the case in batch off-line setting due to lower pure linearity. Thus, we may conclude that a structural change of models is not absolutely necessary in this production process, and therefore an adaptation of parameters sufficient.

Finally, we observed the evolution of residuals and error bars over some splits, an example results for Recipe #4 shown in Figure 4 (residuals as dots, error bars as solid lines around), where the left image shows the behavior of static models and the right one the behavior of dynamically adaptive models. Clearly, we can observe two major results: 1.) the residuals start drifting after a while when applying static models, which is prohibited when permanently updating the models and 2.) the error bars are widening up in case of higher residuals, thus indicating their good correlation with prediction errors. Execution times for model updating and performing prediction on a single sample are significantly beyond real-time for all approaches.

5. Conclusion

In this paper, we presented a new approach for tackling the requirements regarding the on-line prediction of the cloud point in melamine resin production process. The approach introduces a non-linear version of PLS, as latent variables are connected with fuzzy systems, which are adaptable with new incoming on-line data samples investigating recursive parameter updates with structural evolution components. Furthermore, fuzzy systems are connected with a new variant of forward selection, termed as *forward selection with bands (FSB)* which inherits the statistical concepts of forward selection, but is able to extract wavebands with arbitrary size. Additionally, we investigated the implementation of a new model ensemble strategy for noise reduction in re-

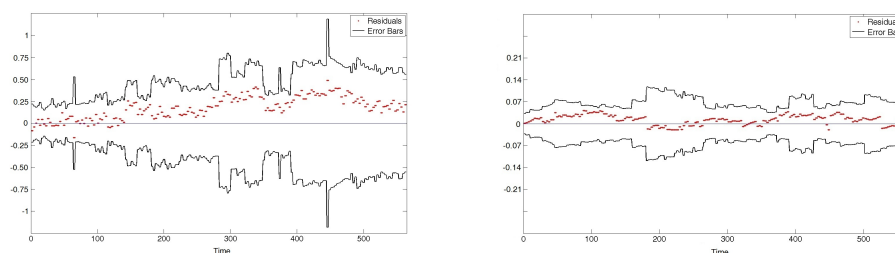


Figure 4: Behavior of residuals (dots) and error bars (surrounding solid lines) for recipe #4, left for static models, right for self-adaptive models; note also the different y-axis scale in both images

peated measurements. The results show that 1.) our approach can outperform state-of-the-art methods during batch off-line modeling cycles and 2.) is able to prohibit residual drifts over time due to permanent model updates.

Acknowledgements

This work was funded by the Austrian research funding association (FFG) under the scope of the COMET programme within the research network 'Process Analytical Chemistry (PAC)' (contract # 825340). This publication reflects only the authors' views.

References

- [1] A.L. Pomerantsev and O.Y. Rodionova. Process analytical technology: a critical view of the chemometricians. *Journal of Chemometrics*, 26(6):299–310, 2012.
- [2] R.G. Brereton. *Chemometrics: Data Analysis for the Laboratory and Chemical Plant*. John Wiley & Sons, Hoboken, New Jersey, 2003.
- [3] J. Workman and L. Weyer. *Practical Guide to Interpretive Near-Infrared Spectroscopy*. CRC Press-Taylor & Francis Group, Boca Raton, Florida, 2007.
- [4] K. Varmuza and P. Filzmoser. *Introduction to Multivariate Statistical Analysis in Chemometrics*. CRC Press, Boca Raton, 2009.
- [5] J.B. Reeves and S.R. Delwiche. Partial least squares regression for analysis of spectroscopic data. *Journal of Near Infrared Spectroscopy*, 11(6):415–431, 2003.
- [6] X. Shao, X. Bian, and W. Cai. An improved boosting partial least squares method for near-infrared spectroscopic quantitative analysis. *Analytica Chimica Acta*, 666(1–2):32–37, 2010.
- [7] C. Cernuda, E. Lughofer, W. Maerzinger, and J. Kasberger. NIR-based quantification of process parameters in polyetheracrylat (PEA) production using flexible non-linear fuzzy systems. *Chemometrics and Intelligent Laboratory Systems*, 109(1):22–33, 2011.
- [8] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15(1):116–132, 1985.
- [9] W. Pedrycz and F. Gomide. *Fuzzy Systems Engineering: Toward Human-Centric Computing*. John Wiley & Sons, Hoboken, New Jersey, 2007.
- [10] A. Miller. *Subset Selection in Regression Second Edition*. Chapman and Hall/CRC, Boca Raton, Florida, 2002.
- [11] Giuseppina Gini, Tushar Garg, and Marco Stefanelli. Ensembling regression models to improve their predictivity: A case study in QSAR (quantitative structure activity relationships) with computational chemometrics. *Appl. Artif. Intell.*, 23(3):261–281, March 2009.
- [12] M. Islam, X. Yao, S. Nirjon, M. Islam, and K. Murase. Bagging and boosting negatively correlated neural networks. *IEEE Transactions on Systems, Man and Cybernetics part B: Cybernetics*, 38(3):771–784, 2008.
- [13] F.L. Minku, A. White, and X. Yao. The impact of diversity on on-line ensemble learning in the presence of concept drift. *IEEE Transactions on Knowledge and Data Engineering*, 22:730–742, 2010.
- [14] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction - Second Edition*. Springer, New York Berlin Heidelberg, 2009.
- [15] J. Gama. *Knowledge Discovery from Data Streams*. Chapman & Hall/CRC, Boca Raton, Florida, 2010.
- [16] E. Lughofer. Single-pass active learning with conflict and ignorance. *Evolving Systems*, 3(4):251–271, 2012.
- [17] E. Lughofer. Flexible evolving fuzzy inference systems from data streams (FLEXFIS++). In M. Sayed-Mouchaweh and E. Lughofer, editors, *Learning in Non-Stationary Environments: Methods and Applications*, pages 205–246. Springer, New York, 2012.
- [18] E. Lughofer. *Evolving Fuzzy Systems — Methodologies, Advanced Concepts and Applications*. Springer, Berlin Heidelberg, 2011.
- [19] L. Ljung. *System Identification: Theory for the User*. Prentice Hall PTR, Prentice Hall Inc., Upper Saddle River, New Jersey, 1999.
- [20] P. Angelov, E. Lughofer, and X. Zhou. Evolving fuzzy classifiers using different model architectures. *Fuzzy Sets and Systems*, 159(23):3160–3182, 2008.
- [21] B.L. Welch. The generalization of 'students' problem when several different population variances are involved. *Biometrika*, 34(1–2):28–35, 1947.
- [22] E. Lughofer and P. Angelov. Handling drifts and shifts in on-line data streams with evolving fuzzy systems. *Applied Soft Computing*, 11(2):2057–2068, 2011.
- [23] P.P. Angelov. Evolving Takagi-Sugeno fuzzy systems from streaming data, eTS+. In P. Angelov, D. Filev, and N. Kasabov, editors, *Evolving Intelligent Systems: Methodology and Applications*, pages 21–50. John Wiley & Sons, New York, 2010.
- [24] M. Pratama, M.J. Er, X. Li, R.J. Oentaryo, E. Lughofer, and I. Arifin. Data driven modeling based on dynamic parsimonious fuzzy neural network. *Neurocomputing*, 110:18–28, 2013.
- [25] T. Næs and H. Martens. Principal component regression in NIR analysis: Viewpoints, background details and selection of components. *Journal of Chemometrics*, 2(2):155–167, 1988.
- [26] N. Kasabov, D. Zhang, and P.S. Pang. Incremental learning in autonomous systems: evolving connectionist systems for on-line image and speech recognition. In *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts*, 2005, pages 120–125, Hsinchu, Taiwan, 2005.
- [27] Z. Bleier, C. Brouillette, and R. Carangelo. A monolithic interferometer for FT-IR spectroscopy. *Spectroscopy*, 14(10):46–49, 1999.
- [28] T. Fearn and A.M.C. Davies. Locally-biased regression. *Journal of Near Infrared Spectroscopy*, 11(6):467–478, 2003.
- [29] T. Hastie, R. Tibshirani, and J. Friedman. Regularized paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1), 2010.