Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Local linear regression for soft-sensor design with application to an industrial deethanizer

Zhanxing Zhu * **Francesco Corona** * **Amaury Lendasse** *
Roberto Baratti ** **Jose A. Romagnoli** ***

* *Dept. of Information and Computer Science, Aalto University School of Science and Technology, Espoo, Finland, (e-mail: {zhanxing.zhu; francesco.corona; amaury.lendasse}@aalto.fi).*
** *Dept. of Chemical Engineering and Materials, University of Cagliari, Cagliari, Italy, (e-mail: baratti@dicm.unica.it)*
*** *Cain Dept. of Chemical Engineering, Louisiana State University, Baton Rouge, LA 70803 USA, (e-mail: jose@lsu.edu)*

Abstract: Soft-sensors for estimating in real-time important quality variables are a key technology in modern process industry. The successful development of a soft-sensor whose performance does not deteriorate with time and changing process characteristics is troublesome and only seldom achieved in real-world setups. The design of soft-sensors based on local regression models is becoming popular. Simplicity of calibration, ability to handle nonlinearities and, most importantly, reduced maintenance costs while retaining the requested accuracy are the major assets. In this paper, we introduce several approaches for defining an appropriate locality neighborhood and we propose a recursive version of local linear regression for soft-sensor design. To support the presentation, we discuss the results in designing a soft-sensor for estimating the ethane concentration from the bottom of a full-scale deethanizer.

Keywords: Process Monitoring, Soft-sensors, Local Linear Regression

1. INTRODUCTION

Estimating in real-time product quality or other important process variables when on-line analyzers are not available is an essential component of modern process industry. Soft-sensors are a key technology for the task and allow to optimize production toward high-quality products while reducing operational and off-specification costs. In chemical and power industry, soft-sensors are extensively used to estimate hard-to-measure primary variables in process units starting from other easy-to-measure secondary variables (Kadlec et al., 2009). Widely accepted technologies for soft-sensor design are based on prediction models like Multivariable Linear Regression (MLR), Principal Component and Partial Least Squares Regression (PCR and PLSR) and Artificial Neural Networks (ANN).

The design of a soft-sensor from data is, however, a daunting process. After a careful selection of relevant secondary variables and representative observations, a prediction model must be selected and its parameters finely tuned. Generally, a single global method is calibrated using all known observations. Even if an accurate soft-sensor is developed, its estimation performances are likely to deteriorate when the process characteristics change. Thus, maintenance issues arise as repeated calibrations are needed to reinstate the performance; in turn, this implies additional money- and time-consuming workloads.

To minimize model maintenance tasks while retaining estimation accuracy, recursive and local methods have been reported in the research and industrial literature. Recursive methods (e.g., Recursive PLS by Qin 1998) can adapt automatically the model to new operating conditions but they are known to function well only with slow changes in process characteristics and, if the global model is linear, only with mild nonlinearities. Local methods (e.g., Lazy Learning by Bontempi et al. 1999) are calibrated only on a small subset of observations in a neighborhood which is similar to the new operating conditions. Local techniques are globally nonlinear, often achieve the requested accuracy and can be promptly upgraded to automatically include new operations. Usually, the local regression model is a simple linear regressor like MLR (or, PCR and PLSR). Such properties make Local Linear Regression (LLR) a valid alternative for soft-sensor design.

Although local methods are classically based on distance and *nearest neighbors* (Cheng and Chiu, 2004) with a fixed (or cross-validated) number of neighbors, how to define similarity (or locality) between observations remains an open issue. Emphasis has been on adjusting the distance metric (Fukunaga and Flick, 1984) and some recent contributions suggested the definition of a *correlation-based* neighborhood specifically designed to learn a linear regression model (Fujiwara et al., 2009, 2010). Without changing the metric, Gupta et al. 2008 proposed adapting the number of neighbors to the local topology of the data by using convex neighborhoods and suggested three strategies like *natural neighbors*, *natural neighbors inclusive* and *enclosing k-nearest neighbors*. Jin et al. 2003 proposed another convex neighborhood based on the *Delaunay tessellation*. In this work, the prime aim is to introduce the

aforementioned strategies for neighborhood definition and discuss their potential in soft-sensor design.

The paper is structured as follows. Section 2 briefly overviews local linear regression and discusses the techniques for neighborhood definition. Section 3 supports the presentation by proposing a recursive version of these local linear regression models and discusses the results on a full-scale problem consisting of estimating the ethane concentration from the bottom of a full-scale deethanizer.

2. LOCAL LINEAR REGRESSION

Local linear regression (LLR) is a nonlinear regressor in the scenario of statistical estimation. The spirit of LLR is that, over a small subset of the input domain, a simple linear regression model can approximate sufficiently well the true mapping to the output. Local linear regression has the property of simplicity of traditional linear regression and it can overcome the drawback of low model accuracy.

Suppose we are given a set of N training samples $\mathcal{X} \to \mathcal{Y} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in \mathcal{R}^d$ and $y_i \in \mathcal{R}$. For an arbitrary input test point $\mathbf{g} \in \mathcal{R}^d$, local linear regression estimates its output as $\hat{y} = \hat{\boldsymbol{\beta}}^T \mathbf{g} + \hat{\beta}_0$, which fits a least-squares hyperplane over the local neighborhood $\mathcal{J}_\mathbf{g}$ of $\mathbf{g}$:

$$(\hat{\boldsymbol{\beta}}, \hat{\beta}_0) = \arg\min_{\boldsymbol{\beta}, \beta_0} \sum_{\mathbf{x}_j \in \mathcal{J}_\mathbf{g}} (y_i - \boldsymbol{\beta}^T \mathbf{g} - \beta_0)^2. \quad (1)$$

The definition of the neighborhood and the number of neighbors are crucial for local linear regression. In this section, we describe several neighborhood definition strategies for local linear regression, from a geometrical point of view.

We divide the different neighborhoods into two categories: *not-enclosing neighborhood* and *enclosing neighborhood*, based on whether the neighborhood $\mathcal{J}_\mathbf{g}$ encloses the test point $\mathbf{g}$. If $\mathcal{J}_\mathbf{g}$ encloses the test point, we call it an enclosing neighborhood; that is, $\mathbf{g} \in \text{conv}(\mathcal{J}_\mathbf{g})$, where the convex hull of a point set $S=\{s_1, \ldots, s_n\}$ is defined as $\text{conv}(S)=\{\sum_{i=1}^n \omega_i s_i | \sum_{i=1}^n \omega_i = 1, \omega_i \geq 0\}$. Intuitively, linear regression over an enclosing neighborhood leads to geometrical interpolation, whereas for not-enclosing neighborhoods we have extrapolation.

2.1 Not-enclosing neighborhoods

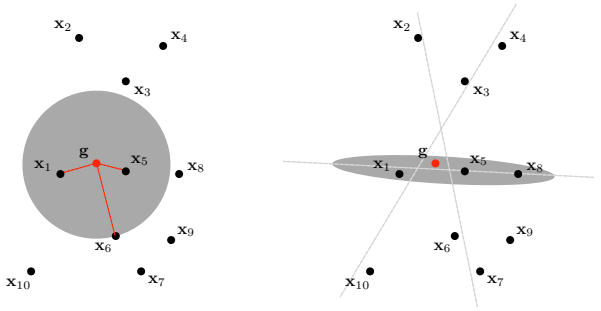
Firstly, we overview two different strategies for defining a not-enclosing neighborhood: classic k-nearest neighbors (kNN) and correlation-based neighborhoods (CoN).

k-nearest neighbors (kNN): Classic k-nearest neighbor defines a neighborhood of $\mathbf{g}$ using k of its neighbors, according to a specified distance metric. Usually, the Euclidean metric is used and the number of neighbors k is fixed or cross-validated. Figure 1(a) shows an example of a kNN neighborhood of size $k = 3$ for the test point $\mathbf{g}$.

Despite its simplicity, one major problem in kNN is the selection of the neighborhood size:

- selecting too few neighbors may lead to a neighborhood that does not enclose the test point (as in Figure 1(a)) which might give a large estimation variance.
- selecting too many neighbors to impose enclosure may cause the regression model to over-smooth.

Thus, how to select adaptively the number k remains an open issue, especially in applications to large scale problems where cross-validation is computationally unbearable.



(a) k-nearest neighbors (b) Correlation-based neighbors

Fig. 1. Not-enclosing neighborhoods: k-nearest neighbors $\mathcal{J}_\mathbf{g}^{kNN}=\{\mathbf{x}_1, \mathbf{x}_5, \mathbf{x}_6\}$ and correlation-based neighbors $\mathcal{J}_\mathbf{g}^{CoN}=\{\mathbf{x}_1, \mathbf{x}_5, \mathbf{x}_8\}$.

Correlation-based neighbors (CoN): The correlation-based neighborhood proposed by Fujiwara et al. (2009) tries to find neighbors that are most correlated with the current test point. The J-statistic by Raich and Cinar (1994), a combination of T^2 and Q statistics from Principal Components Analysis (PCA), is used as index of correlation dissimilarity. In the example of Figure 1(b), points $\{\mathbf{x}_1, \mathbf{x}_5, \mathbf{x}_8\}$ are found as the most correlated with $\mathbf{g}$ and, thus, define its neighborhood. The basic strategy has several parameters to be tuned but a promising extension based on spectral clustering has been recently proposed by the same authors (Fujiwara et al., 2010). Unfortunately, the computational complexity of the extension is at least $\mathcal{O}(N^3)$ making it unusable in many practical applications.

In soft-sensor design, Correlation-based Just-in-Time (Co-JIT) uses a conventional CoN with Just-in-Time modeling, JIT (Bontempi et al., 1999). A moving window with fixed length is used to create a sequence of sample sets and the most correlated sample set is chosen for LLR modeling.

2.2 Enclosing neighborhoods

Recently, Gupta et al. (2008) proved that if a test point is in the convex hull enclosing its neighborhood, then the variance of the local linear regression estimate is bounded by the variance of the measurement noise. Such a property is fundamental to avoid erratic results. The authors also suggested three enclosing neighborhood definition strategies; these strategies are overviewed in the following.

Enclosing k-nearest neighbors (ekNN): It is based on the kNN of the test point $\mathbf{g}$ and extends it to define a neighborhood that encloses it, Figure 2(a). ekNN is the neighborhood of the kNNs with the smallest k such that $\mathbf{g} \in \text{conv}(\mathcal{J}_\mathbf{g}(k))$, where $\mathcal{J}_\mathbf{g}(k)$ is the set of kNNs of $\mathbf{g}$ (Gupta et al., 2008). If $\mathbf{g}$ is outside of convex hull of the set $\mathcal{X}$, no such k exists. Define *distance to enclosure* as

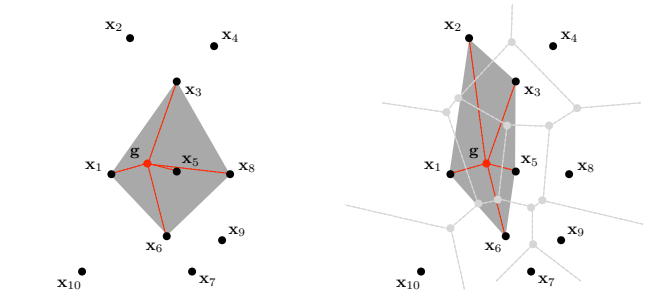
$$D(\mathbf{g}, \mathcal{J}_\mathbf{g}) = \min_{\mathbf{z} \in \text{conv}(\mathcal{J}_\mathbf{g})} \|\mathbf{g} - \mathbf{z}\|_2, \quad (2)$$

where $\mathbf{z}$ is any point in the convex hull around the neighborhood of $\mathbf{g}$. Note that $D(\mathbf{g}, \mathcal{J}_\mathbf{g}) = 0$ only if $\mathbf{g} \in \text{conv}(\mathcal{J}_\mathbf{g})$. Then, the e$k$NN neighborhood is $\mathcal{J}_\mathbf{g}(k^*)$ with

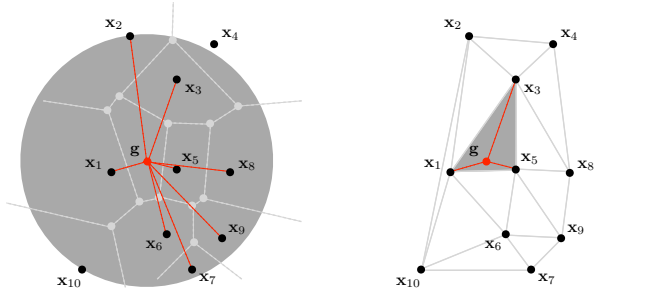
$$k^* = \min_k \{k | D(\mathbf{g}, \mathcal{J}_\mathbf{g}(k)) = D(\mathbf{g}, \mathcal{X})\}. \quad (3)$$

The computational complexity for building a convex hull using k neighbors is $\mathcal{O}(k^{\lfloor d/2 \rfloor})$, where $\lfloor \cdot \rfloor$ is a floor function.

Natural neighbors (NN): Natural neighbors are based on the Voronoi tessellation of the training samples and the test point (Sibson, 1981). The natural neighbors of $\mathbf{g}$ are defined as those points whose Voronoi cells are adjacent to the cell including $\mathbf{g}$. Natural neighbors has the so-called *local coordinates property*, which is used to prove that the natural neighbors form an enclosing neighborhood if $\mathbf{g} \in \text{conv}(\mathcal{X})$. Figure 2(b) shows an example of natural neighbors.



(a) Enclosing k-nearest neighbors (b) Natural neighbors



(c) Natural neighbors inclusive (d) Delaunay neighbors

Fig. 2. Enclosing neighborhoods: Enclosing k-nearest neighbors $\mathcal{J}_\mathbf{g}^{eKNN}=\{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_8\}$, Natural neighbors $\mathcal{J}_\mathbf{g}^{NN}=\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_5, \mathbf{x}_6\}$, Natural neighbors inclusive $\mathcal{J}_\mathbf{g}^{NNi}=\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9\}$ and Delaunay tessellation neighbors $\mathcal{J}_\mathbf{g}^{DTN}=\{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_5\}$.

Natural neighbors inclusive (NNi): In some cases of non-uniformly distributed local areas, a training point which is far from the test point can be one of its natural neighbors, but a nearer point is excluded for its neighborhood. In order to overcome this situation, natural neighbors inclusive has been proposed (Gupta et al., 2008) to include both the natural neighbors and those training points within the distance to the furthest natural neighbor. That is,

$$\mathcal{J}_\mathbf{g}^{NNi} = \{\mathbf{x}_j \in \mathcal{X} | \|\mathbf{g} - \mathbf{x}_j\| \leq \max_{\mathbf{x}_i \in \mathcal{J}_\mathbf{g}^{NN}} \|\mathbf{g} - \mathbf{x}_i\|\}. \quad (4)$$

Figure 2(c) is an example of natural neighbors inclusive.

Delaunay tessellation neighbors (DTN) and Delaunay topological regression (DTR): Proposed by Jin et al. (2003), the strategy is based on the Delaunay triangulation (or tessellation in a d-dimensional space, $d > 2$) of the training points. After tessellation, a neighborhood is defined from the vertices of the triangle (or polyhedron) that envelopes the test point $\mathbf{g}$. Figure 2(d) shows an example.

Based on the DTN of the test point, Delaunay topological regression does not work according to Equation 1, for it estimates the output through interpolation. The output of $\mathbf{g}$ is estimated as a convex combination of the outputs $\mathbf{y} = (y_1 \quad \cdots \quad y_{d+1})^T$ of the vertices of its enclosing:

$$\hat{y} = \boldsymbol{\alpha}^T \begin{pmatrix} y_1 \\ \vdots \\ y_{d+1} \end{pmatrix}, \text{ with } \sum_{i=1}^{d+1} \alpha_i = 1 \quad \text{and} \quad \alpha_i \geq 0. \quad (5)$$

The weight vector $\boldsymbol{\alpha}$ is calculated from the linear system

$$\boldsymbol{\alpha} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_{d+1} \\ 1 & 1 & \cdots & 1 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{g} \\ 1 \end{pmatrix}, \quad (6)$$

where $\mathbf{x}_1, \cdots, \mathbf{x}_{d+1}$ are the vertices of the enclosing of $\mathbf{g}$. If the test point does not fall inside the convex hull defined by known data points, it is not even inside any constructed polyhedron. To overcome this situation, there are several approaches reported in the literature. In our experiments for local linear regression in soft-sensor design, we prefer the stability of the projection method proposed by Corona et al. (2010) over the methods reported by Jin et al. (2003). The method constructs a consistent estimate for the external data points by searching for their closest projection onto the convex hull. As the projected points are on the facet of the convex hull, their weights and output estimates can be calculated using Equation 6 and 5.

3. THE INDUSTRIAL DEETHANIZER

In this section, the development of a soft-sensor using the strategies for local linear regression previously introduced is discussed on a full-scale deethanizer. The recursive extensions of the methods are presented and the results compared and discussed.

The deethanizer, Figure 3, separates ethane from a feed stream of light naphtha with the operational objective to produce as much ethane as possible; that is, operations should minimize propane's concentration in the top while satisfying a constraint on the amount of ethane in the bottom. Such a constraint is quantified by the maximum concentration of ethane lost from the bottom; the operation range is set to be within 1.8−2%. The other constraint is on the maximum concentration of propane from the top, which is set to be smaller than 2%. Breaking the constraints has important economic implications: Out-of-specification products (high ethane and/or propane concentrations) and unnecessary production costs (low ethane and /or propane concentrations). According to the plant's management, the constraint on the bottom ethane is rarely met. In this work, we focus on estimating this variable.

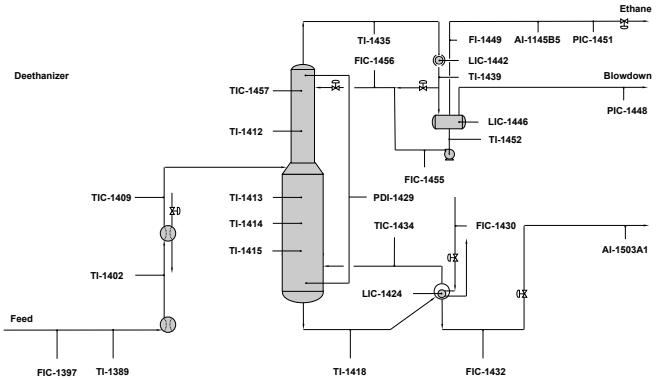


Fig. 3. The deethanizer: Flowsheet with instrumentation.

Although the ethane concentration is analyzed by a continuos-flow chromatograph, a measure is returned only every 18 minutes and with a delay of 90 minutes. Clearly, the delay and low sampling frequency associated with the analytical measurement of ethane pose severe limitations to the integration of the analysis within a control strategy. Hence, the goal is to develop a soft sensor capable to estimate in real-time this primary variable, starting from a set of secondary process variables measured online.

3.1 Sensor development

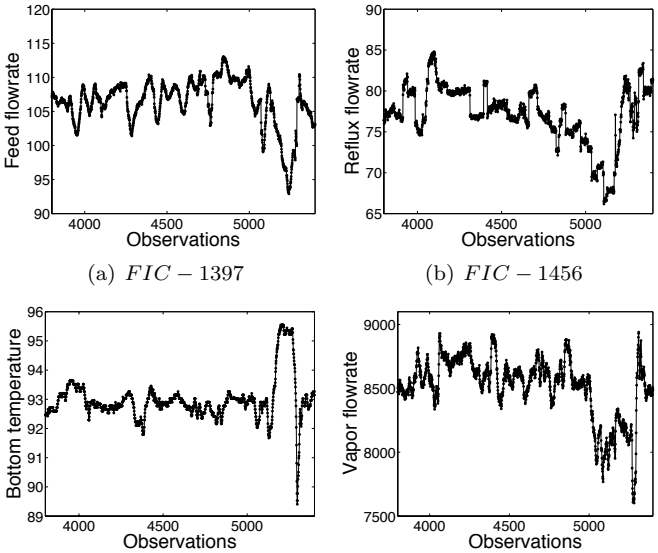
For the scope, a set of process measurements has been collected from the plant's distributed control system. The data correspond to 3 weeks of continuous operation in winter asset. The output variable of the soft-sensor is the ethane concentration AI-1503$A1$ (Figure 3). Although the data are available as 3-minute averages for 26 process variables, for the analysis only 9 have been retained as inputs to the soft-sensor (Table 1). The selection is based on the physical knowledge of the process (Corona et al., 2009). The number of available input observations is 7200 and the number of output observations is 1200, because measured every 6 sampling times.

Table 1. The deethanizer: Selected input variables for the soft-sensor.

Variable (TAG)	Variable (TAG)
Feed Flowrate (FIC-1397)	Vapor Flowrate (FIC-1430)
Enriching Temp. (TIC-1457)	Vapor Temp. (TIC-1434)
Reflux Flowrate (FIC-1456)	Bottom Temp. (TI-1414)
Reflux Temp. (TI-1452)	Bottom Temp. (TI-1418)
Distillate Pressure (PIC-1451)	

Initially, to set a reference for comparison, a PLSR model and a Local Linear Regression model over a kNN neighborhood (LLR-kNN) has been calibrated using the first one-third of the available data (400 observations, where both the input and the output variables are measured). The remaining two-thirds of the data (800 observations) have been used as independent test for the models. The number of latent variables (2) used in this static PLSR model has been optimized by standard Leave-One-Out cross-validation, LOO (Hastie et al., 2009). As for static LLR-kNN model, the number of nearest neighbors has been fixed to be equal to 10% of the calibration set. Given the presence of collinearity among the inputs, the dimensionality of the input space has been reduced to 2 using a Principal Component Analysis. Also the number of components in the PCA model has been cross-validated.

To present the estimation performances, a testing period corresponding to a 3-day window is reported. This window has been selected because, during the testing period, the unit was subjected to an abrupt feed change (in flowrate and possibly in composition) and run under critical operating conditions. The temporal evolution of 4 relevant process variables is depicted in Figure 4; here, it is possible to note how the variation in the feed triggered an action on the vapor to reboiler flowrate and in the reflux flowrate.

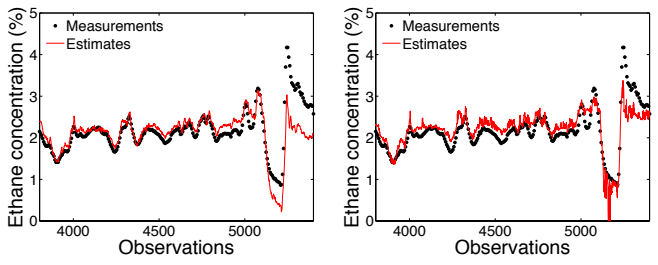


(a) $FIC-1397$ (b) $FIC-1456$

(c) $TI-1418$ (d) $FIC-1430$

Fig. 4. Operation measurements for a selection of inputs.

The events initiated a sequence of oscillations in the ethane concentration which has been only partially recovered by the static sensors (Figure 5). In particular, it is possible to notice that the static PLSR is not capable to recover the full extension of the variation, and the static LLR-kNN is unstable and it also lacks an overall accuracy during the normal operation of the column. Over the full testing period, the accuracy of the static models expressed in terms of root mean squared error (RMSE) is equal to 0.311% for PLSR and 0.358% for LLR-kNN. The RMSE is used as figure of merit to assess accuracy because it is expressed in the same units of the output variable (%).



(a) Static PLSR (b) Static LLR-kNN

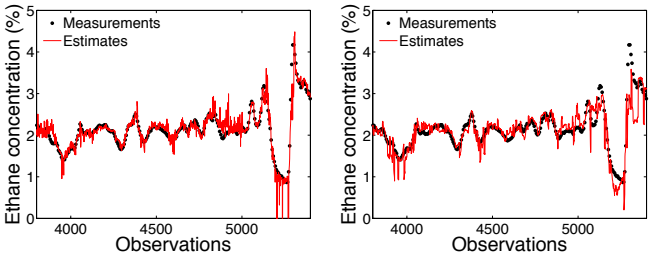
Fig. 5. Estimation results for the static models.

The results obtained using static models motivate the idea of applying local linear regression recursively for the development of a more accurate soft-sensor while trying to minimize the maintenance tasks.

Recursive models: In the recursive design, the basic principle of JIT modeling with a moving window to upgrade the models is used. Concretely, a moving window with fixed length W is first created. Within the current window, when a new input measurement is ready and an estimation is needed, a local regression model is constructed. Then, the output of this new input is estimated with the current model. Whenever a new input-output measurement is available, it is added to the window and the oldest observation dropped out to keep a fixed length. This simple principle can be used with local linear regression methods over kNN, NN, NNi, ekNN neighborhoods and DTR.

To avoid collinearity and impose orthogonality between the inputs, PCA can be used for dimensionality reduction in each moving window (recursive local PCR). For comparison, a number of components ranging from 2 to 5 has been chosen. The other parameter of the method is the length of the moving window; $W = 50$ has been used in the experiments (i.e., up to 50 input-output observations in each window are available for training the models).

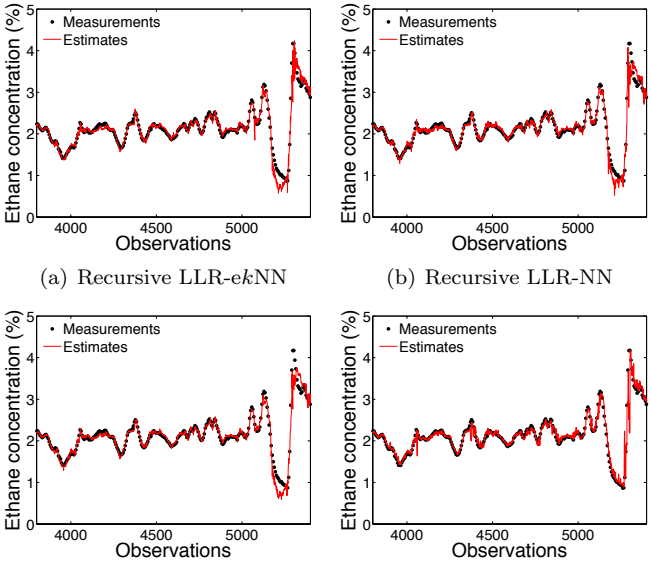
The estimation results over the discussed test window are depicted in Figure 6 for recursive LLR-kNN and for correlation-based JIT. Notice that for the kNN neighborhood, the number of nearest neighbors had to be selected beforehand; in the experiments, the number of nearest neighbors is fixed and set to be equal to 10% of the window length W, $k = 0.1W$. On the other hand, CoJIT works differently as it updates the model on the basis of a correlation similarity between the chosen data sets and the current test point, see (Fujiwara et al., 2009) for details. The parameters of CoJIT has been set as i) combination coefficient of J statistic $\lambda = 0.01$, ii) threshold of J statistic $J_I = 0$, iii) number of principal components varying between 2 and 5 and iv) window length $W \in \{10, 20, \ldots, 50\}$.



(a) Recursive LLR-kNN (b) CoJIT

Fig. 6. Estimation results for the recursive local linear methods with not enclosing neighborhoods.

In terms of accuracy, the Recursive LLR-kNN achieved a RMSE equal to 0.373% (using 2 principal components) and CoJIT a RMSE equal to 0.312% (using 3 principal components and a window-length $W = 50$). From the figure, it is possible to notice how the use of a recursive of LLR-kNN over a static one is beneficial only at reducing the offset in the estimates. However, the smaller number of points available for training the model leads to instability. As pointed out in Section 2, the effect is due to a not-enclosing neighborhood. Although based on a different principle, also CoJIT suffers from the same limitation.



(a) Recursive LLR-ekNN (b) Recursive LLR-NN

(c) Recursive LLR-NNi (d) Recursive DTR

Fig. 7. Estimation results for the local linear methods with enclosing neighborhoods.

The shortcomings associated with an unbounded variance of the estimates are overcome by using the enclosing neighborhoods. Figure 7 shows the results over the discussed testing window for the recursive versions of LLR-ekNN, LLR-NN and LLR-NNi. From the diagrams, it is possible to notice how the estimates strongly benefitted in terms of accuracy and stability. Also the dynamic changes associated with the process operations are promptly and correctly recovered although a tendency to over-estimate the full magnitude of the variation is observed (an effect already observable with the previously discussed models).

Figure 7(d) shows the results for the recursive DTR model which, instead of fitting a local linear model, performs interpolation within the enclosing. On the discussed testing window, the method performs in a rather accurate way and fully recovers the dynamic variations in the unit. However, it is also possible to notice the presence of a few unstable predictions; a behavior easily explained by recalling that in a DTR model the size of the enclosing neighborhood is minimal given the dimensionality of the input space.

3.2 Discussion

Table 2 presents the estimation accuracy for all the recursive methods used in the experiments. The table clearly summarizes the benefit of using an enclosing neighborhood over a not-enclosing one. In addition, it is important to notice that such neighborhoods are characterized by an intrinsic adaptivity in constraining locality, which allows for the design of a regression model which is virtually parameter-free. When used in soft-sensor design, the only parameters that need to be tuned are related to intrinsic complexity of the function that is to be estimated (via the selection of input variables or, if dimensionality reduction is used, the number of variables to be retained) and the

dynamics of the process to be modeled (via the definition of the length of the moving window).

Table 2. Estimation results as $RMSE$ in prediction. The underlined RMSEs corresponds to the models reported in the figures.

PCs	kNN	CoJIT	NN	NNi	ekNN	DTR
2	0.373	0.319 (50)	0.335	0.285	0.757	0.267
3	0.515	0.315 (40)	0.283	0.261	0.257	0.246
4	3.584	0.312 (50)	0.251	0.246	0.231	0.243
5	2.760	0.330 (50)	0.259	0.253	0.254	0.244

For the sake of completeness, we also report that a recursive PLSR model with the same window length and 3 latent variables was able to outperform all the recursive local linear models presented. The test accuracy of this Recursive-PLSR model in terms of RMSE is 0.180%, confirming the unquestionable quality of this method.

4. CONCLUSION

In this work, the design of soft-sensors based on local linear regression has been investigated. Several strategies for defining locality have been introduced and the potentialities for developing a recursive version of the methods have been proposed and illustrated on a real-world application.

The experimental results achieved on estimating the ethane concentration in a full-scale industrial deethanizer, confirmed that an appropriate selection of the neighborhood is critical for learning a local regression model. Enclosing neighborhoods are preferable, mostly because of their stability but also due to their automatic adaptivity to the topology of the measurements, which eliminates the necessity of user-defined parameters. In this respect, the recursive extensions of the methods are virtually parameter-free, if we exclude the length of the moving window and the definition of the input space dimensionality; such parameters are, however, problem related and can be learned from data. Among the local regression methods, the best accuracies have been achieved using an enclosing kNN neighborhood and a Delaunay tessellation. Overall, the methods achieved an accuracy which is comparable to the standard methods used in industry and the analytical accuracy of the measurements.

As a final remark, we would like to point out an important aspect that needs to be taken into account when implementing a soft-sensor based on a recursive local linear method. A successful application of the techniques in a real-world scenario is possible only if the developed sensor is complemented with an appropriate validation system on the process measurements. Concretely, if the soft-sensors are used to support existing on-line measurements, the analytical instruments need to be carefully maintained in order to avoid a fit against unreliable measurements. In the presence of unreliable measurements, the calibration would inevitably lead to undesirable results like undetectable biases and drifts in the analysis, as well as faulty hardware sensors. The same holds also when the soft-sensors are calibrated against laboratory measurements.

ACKNOWLEDGMENTS

Jose A. Romagnoli and Francesco Corona gratefully thank the Regione Sardegna for the support through the program *Visiting Professor 2010*.

REFERENCES

Bontempi, G., Birattari, G., and Bersini, M. (1999). Lazy learning for local modeling and control design. *International Journal of Control*, 72, 179–186.

Cheng, C. and Chiu, M.S. (2004). Nonlinear process monitoring using JITL-PCA. *Chemometrics and Intelligent Laboratory Systems*, 76, 1–13.

Corona, F., Liitiäinen, E., Lendasse, A., Baratti, R., and Sassu, L. (2010). A continuous regression function for the delaunay calibration method. In *Proceedings of IFAC/DYCOPS 2010 9th International Symposium on Dynamics and Control of Process Systems*, 180–185. Leuven, Belgium.

Corona, F., Mulas, M., Baratti, R., and Romagnoli, J.A. (2009). Data derived analysis and inference for an industrial deethanizer. In *Proceedings of IFAC/ADCHEM 2009 7th International Symposium on Advanced Control of Chemical Processes*, 717–723. Instanbul, Turkey.

Fujiwara, K., Kano, M., and Hasebe, S. (2010). Development of correlation-based clustering method and its application to software sensing. *Chemometrics and Intelligent Laboratory Systems*, 101, 130–138.

Fujiwara, K., Kano, M., Hasebe, S., and Takinami, A. (2009). Soft-sensor development using correlation-based just-in-time modeling. *American Institute of Chemical Engineers Journal*, 55, 1754–1765.

Fukunaga, K. and Flick, T. (1984). An optimal global nearest neighbor metric. *IEEE Transactions on Patter Analysis and Machine Intelligence*, PAMI-6, 314–318.

Gupta, M.R., Garcia, E.K., and Chin, E. (2008). Adaptive local linear regression with application to printer color management. *IEEE Transactions on Image Processing*, 17, 936–945.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *Elements of Statistical Learning: Second Edition*. Springer, New York.

Jin, L., Fernández Pierna, J.A., Xu, Q., Wahl, F., de Noord, O.E., Saby, C.A., and Massart, D.L. (2003). Delaunay calibration method for multivariate calibration. *Analytica Chimica Acta*, 488, 1–14.

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

Qin, S.J. (1998). Recursive PLS algorithms for adaptive data modeling. *Computers and Chemical Engineering*, 22, 503–514.

Raich, A. and Cinar, A. (1994). Statistical process monitoring and disturbance diagnosis in multivariable continuous processes. *American Institute of Chemical Engineers Journal*, 42, 995–1009.

Sibson, R. (1981). A brief description of natural neighbors interpolation. In V. Barnett (ed.), *Interpreting Multivariate Data*, 21–36. Wiley, New York.