Soft Sensor based on adaptive local learning

Petr Kadlec and Bogdan Gabrys

Computational Intelligence Research Group, Smart Technology Research Centre, Bournemouth University, Fern Barrow, Poole, BH12 5BB, United Kingdom
pkadlec@bournemouth.ac.uk

Abstract. When it comes to application of computational learning techniques in practical scenarios, like for example adaptive inferential control, it is often difficult to apply the state-of-the-art techniques in a straight forward manner and usually some effort has to be dedicated to tuning either the data, in a form of data pre-processing, or the modelling techniques, in form of optimal parameter search or modification of the training algorithm. In this work we present a robust approach to on-line predictive modelling which is focusing on dealing with challenges like noisy data, data outliers and in particular drifting data which are often present in industrial data sets. The approach is based on the local learning approach, where models of limited complexity focus on partitions of the input space and on an ensemble building technique which combines the predictions of the particular local models into the final predicted value. Furthermore, the technique provides the means for on-line adaptation and can thus be deployed in a dynamic environment which is demonstrated in this work in terms of an application of the presented approach to a raw industrial data set exhibiting drifting data, outliers, missing values and measurement noise.

1 Introduction

In this work we propose an adaptive regression model for on-line prediction. The predictor is applied in the process industry where this type of model is called data-driven *Soft Sensor* [1]. Data-driven Soft Sensors are trained using *historical process data*. For many industrial applications this data is automatically recorded and can be retrieved from the process databases. After the training, the models are launched in the real-life application where it is the task of the Soft Sensor to provide an on-line prediction of the target values. The on-line data stream is often unlabelled or the labels are delayed (e.g. due to manual evaluations in chemical laboratories). In the latter case the delayed labels can be used for the adaptation of the models.

The most common data-driven techniques applied for empirical Soft Sensor modelling are: (i) Principle Component Regression [2] and Partial Least Squares method [3], these techniques gained their popularity due to their statistical background, ease of interpretability of the model, and due to the fact that the methods intrinsically deal with data co-linearity which is a common problem among industrial data sets. Examples of Soft Sensor applications based on PCA/PLS are [4, 5]; (ii) Artificial Neural Networks [6], this modelling technique is very popular in predictive modelling and due to its ability to model non-linear problems it has also found broad applications as Soft Sensor

modelling tool (see e.g. [7]); (iii) several other methods like Support Vector Machines [8] and Neuro-Fuzzy Systems [9] for on-line prediction [10, 11].

This work is based on the local learning approach. The pioneering work of local learning was presented by Jacobs and Jordan in [12], where a gating network is used to decide which model from a set of available local experts is responsible for the prediction of the given input sample. Another approach related to our work is the Locally Weighted Learning (LWL) technique [13] which was proposed as an effective way of dealing with the bias/variance dilemma [14] and negative interference [15]. Based on LWL the Locally Weighted Projection Regression has been proposed in [16] where additionally a local dimensionality reduction has been applied in order to deal with high dimensional data with locally low dimensional manifold. Furthermore, the LWPR algorithm is restricted to the application of linear regression models as predictors. In contrast to LWPR the approach presented in this work provides higher flexibility by modelling the receptive fields using the Parzen method for distribution approximation [17] and allows to use any regression technique to be applied as local expert. Also, the receptive field descriptors are built in the reduced low dimensional space rather than in the original high dimensional input space.

2 Soft Sensor for on-line prediction

This section presents the main contribution of this work which is a Soft Sensor algorithm. The algorithm can be split into the following steps: (i) Receptive fields construction; (ii) Local models/experts training; (iii) Receptive field descriptors building (iv) Local experts combination; (v) Soft Sensor adaptation during the run-time. The first three steps are performed off-line using the historical data while the last two steps are carried out during the exploitation (run-time) phase and thus using the on-line data.

Receptive fields construction: The aim of the receptive field building is to divide the historical data into partitions representing the particular concepts within the historical data. The notion of the concepts is linked to the area where a model, called landmarker, provides constant performance.

Provided the historical data set $S^{hist} = (\mathcal{X}^{hist}, \mathcal{Y}^{hist})$, where S^{hist} consists of an $M \times N$ matrix of input data $\mathcal{X}^{hist}$, consisting of M variables and N observations, and a vector of N target labels $\mathcal{Y}^{hist}$, the first step of the algorithm is training the landmarker. The landmarker is trained using samples from an initial window S^{init} which is a subset of the historical data:

$$S_i^{init} = (\mathcal{X}_i^{init}, \mathcal{Y}_i^{init}) := \{(\mathbf{x}_n, y_n); n = k, \ldots, k + l\}\,, \tag{1}$$

where i identifies the current receptive field, n is the index of the samples within the receptive field, k is the index of the first sample in the current receptive field, l is the length of initial window, $\mathbf{x}_n$ is M-dimensional input sample and y_n is the corresponding target value.

Provided the initial set, the landmarker f_i^{lm} can be trained and the residual vector $\mathbf{r}_i^{init}$ of the trained landmarker on the training data can be calculated:

$$\mathbf{r}_i^{init} = \mathcal{Y}_i^{init} - f_i^{lm}(\mathcal{X}_i^{init}), \tag{2}$$

The next step is shifting the window one step forwards, while keeping the size of window constant:

$$S^{shifted} = (\mathcal{X}_i^{shifted}, \mathcal{Y}_i^{shifted}) := \{(\mathbf{x}_n, y_n); n = k+s, \ldots, k+s+l\} \quad s = 1, \tag{3}$$

and calculating the new residual values $\mathbf{r}_i^{shifted}(s)$ of landmarker's prediction using the new data window.

As next the two residual vectors ($\mathbf{r}_i^{init}$ and $\mathbf{r}_i^{shifted}(s)$) are tested for a statistically significant difference using the one-sided t-test [18]. This significance test was chosen because the residuals can be, ideally, assumed as normally distributed. The null hypothesis is that there is no significant difference in the mean values of the two samples which in this case means that there is no significant difference between the landmarker's performance on the initial window and on the shifted window, i.e. the data from S^{init} and $S^{shifted}$ belong to the same concept. This procedure is repeated as long as the null hypothesis of the significance test remains valid:

$$s^{final} = \underset{s \in [1,\ldots,N-k]}{\operatorname{argmin}} (ttest(\mathbf{r}_i^{init}, \mathbf{r}_i^{shifted}(s)) == 1), \tag{4}$$

where s^{final} corresponds to the first sample for which the t-test rejects the null hypothesis.

Finally, the receptive field is constructed in the following way:

$$S_i^{RF} = (\mathcal{X}_i^{RF}, \mathcal{Y}_i^{RF}) := \left\{(\mathbf{x}_n, y_n); n = k, \ldots, k+l+s^{final}-1\right\} \tag{5}$$

and the algorithm can move to the next receptive field by constructing new initial window S_{i+1}^{init}.

The outcome of this stage is a set of receptive fields $\mathcal{S}^{RF}$, each corresponding to a concept of the historical data.

Local experts training: After identifying of the receptive fields, a local expert f^{le} can be trained for each of the receptive fields. At this point one can apply any computational learning technique for the local experts with the same or higher than the landmarker's model complexity. This constrain originates from the fact that the notion of the concept is related to the landmarker's prediction capabilities and thus in order to be able to model the concept using the local expert it has to posses the same or better prediction ability.

After this step there is a set of trained local experts $\mathcal{F}^{le} := \left\{(f_i^{le}); i = 1, \ldots, I\right\}$.

Receptive field descriptors building: The next step towards the final model is building descriptors of the receptive fields which will be later sampled to estimate the correspondence of the test samples to the particular receptive fields.

The aim of the descriptors is to describe the area of expertise of the particular local experts. This task is approached by building a set of two-dimensional descriptors, which are a combination of the input variables distribution $D(\mathcal{X}_i^m)$ and of the distribution of the target variable $D(\mathcal{Y}_i)$ within the ith receptive field. This leads to a two-dimensional probability density function $D_i^m(\mathcal{X}_i^m, \mathcal{Y}_i)$. The mutual descriptor $D_i^m(\mathcal{X}_i^m, \mathcal{Y}_i)$ is constructed using a two-dimensional Parzen method [17].

The final descriptor $\mathcal{D} := \{D_i^m\}$ is a set of $M \times I$ two-dimensional pdfs, with M being the number of input variables and I the number of receptive fields.

Local experts combination: During the run-time (on-line) phase the task of the Soft Sensor is to make an on-line prediction of the target variable given the unlabelled test samples. To obtain the final prediction y^{final} these predictions have to be combined. This is done by making a weighted sum of the local experts' predictions:

$$y^{final} = \sum_{i=1}^{I} v_i(\mathbf{x}, f_i^{le}) f_i^{le}(\mathbf{x}), \tag{6}$$

where $v_i(\mathbf{x}, f_i^{le})$ is the weight of the ith local expert's prediction and $f_i^{le}(\mathbf{x})$ is the actual prediction given the input vector $\mathbf{x}$. In the case of the presented approach the purpose of the weights is to predict the performance of the local expert given the input sample and the local expert's prediction and thus to select the correct local expert for making the prediction. This can be expressed as the posterior probability of the ith receptive field given the test sample $\mathbf{x}$ and the local expert prediction $f_i^{le}(\mathbf{x})$:

$$v_i(\mathbf{x}, f_i^{le}) = p(i|\mathbf{x}, f_i^{le}) = \frac{p(\mathbf{x}, f_i^{le}|i)p(i)}{p(\mathbf{x}), , f_i^{le}}, \tag{7}$$

where $p(i)$ is the a priori probability of the ith receptive field, $p(\mathbf{x}, f_i^{le})$ is the normalisation factor to normalise $p(i|\mathbf{x}, f_i^{le})$ and $p(\mathbf{x}|i)$ the likelihood of $\mathbf{x}$ given the receptive field and f_i^{le} which can be calculated by reading the mutual pdfs at the positions defined by the sample $\mathbf{x}$:

$$p(\mathbf{x}, f_i^{le}|i) = \prod_{m=1}^{M} p(x^m, f_i^{le}|i) = \prod_{m=1}^{M} D_i^m(x^m, f_i^{le}(\mathbf{x}))p(m), \tag{8}$$

with $p(m)$ as a priori probability for the different variables calculated as its entropy:

$$p(m) = -\sum_{m=1}^{M} D_i^m(x^m, f_i^{le}) \log(D_i^m(x^m, f_i^{le})), \tag{9}$$

in this way the discriminant power of the variables is measured and used as a weighting factor.

Soft Sensor Adaptation: The adaptation is performed at the level of the combination of the local experts while the actual local experts are kept untouched. The adaptation is done by modifying the receptive field descriptors.

The descriptors are modified each time a correct target value y_n is received. Having the correct target value, the error e_i of the ith local expert's prediction f_i^{le} can be calculated and normalised:

$$e_i = (y_n - f_i^{le}(\mathbf{x}_n))^2, e_i^{rel} = \frac{e_i}{\sum_{i=1}^{I} e_i}. \tag{10}$$

The relative error values are further on applied as weighting factor for the adaptation masks ΔD_i^m. These are two-dimensional Gaussian functions which are used to adapt the neighbourhood of the sampling point $(x_n^m, f_i^{le}(\mathbf{x}_n))$ of the descriptors D_i^m:

$$\Delta D_i^m(x,y) = \alpha e_i^{rel}((\Delta D_i^m(x))^T * \Delta D_i^m(y)) \tag{11a}$$

$$\Delta D_i^m(x) = \frac{1}{h^x\sqrt{2\pi}} \exp(-\frac{(\mathbf{x}^{range} - x_n^m)^2}{2(h^x)^2}) \tag{11b}$$

$$\Delta D_i(y) = \frac{1}{h^y\sqrt{2\pi}} \exp(-\frac{(\mathbf{y}^{range} - y_n)^2}{2(h^y)^2}), \tag{11c}$$

where $\Delta D_i^m(x,y)$ is a two-dimensional adaptation mask for the m variable and ith receptive field, $\Delta D_i^m(x)$ and $\Delta D_i(y)$ are the marginal masks for the mth input variable and the target variable respectively, α is the adaptation strength parameter, which can be used for modifying the power of the adaptation, further on h^x is the variance, i.e. the width, of the input variable Gaussian kernel and h^y the variance of the target variable kernel respectively, these two values define the size of the neighbourhood of the current sampling point which is being modified by the adaptation mask ΔD_i^m. This parameter influences the adaptation properties and can be tuned in order to obtain optimal adaptation capability, and $\mathbf{x}^{range}$ and $\mathbf{y}^{range}$ are the ranges within which the descriptor D_i^m stores the distributions.

Finally, the descriptors can be adapted using the modification masks in the following way:

$$D_i^m(t+1) = D_i^m(t) \cdot \Delta D_i^m, \tag{12}$$

where $\cdot$ is the Hadamard product.

3 Experiments

In this section the proposed adaptive Soft Sensor will be evaluated in terms of an industrial data set. The raw data set provided by Evonik Industries consists of 1066 raw samples (covering six months of the operation of the process) each having 19 input features, like temperatures and pressures measured within the process plant. The target values of this data set are laboratory measurements of the residual humidity of the process product. The data exhibit typical problems of industrial data like missing values, outliers, measurements noise.

The available data is split into two parts. The first 30% of the samples form the historical data which is used for training of the three models. The remaining 70% of the data simulate the on-line data and are used as on-line testing data. This split was chosen because the focus of this work is on the assessment of the adaptivity capabilites of the Soft Sensor. All the tests presented in this section are strictly following the out-of-sample principle, i.e. the models are tested on unseen test samples.

Based on the analysis of the models' sensitivity to the input parameter(s), following values of the input parameters for the non-adaptive/adaptive local Soft Sensor have been selected: initial window size $l = 50$ (see Eq. 1), adaptation rate $\alpha = 1000$ (see Eq. 11a). The applied landmarkers are multiple linear regression models and the local expert are

committee of ten multiple regression models combined using the LMS method. Prior to the training of the local experts the data are locally pre-processed using the Principle Component Analysis (PCA) with eight principle components (the optimal number of principle components was obtained using the cross-validation approach). There are two version of the proposed models discussed, a non-adaptive version without any adaptation capabilities and an adaptive version based on the adaptation of the receptive field descriptors.

Additionally for performance comparison, the non-adaptive version of the proposed model is compared to a traditional Soft Sensor, namely an Multi-Layer Perceptron ANN. The MLP Soft Sensor is a committee of ten randomly initialised models each having five hidden units in one hidden layer. These parameters were found as optimal in terms of Mean Squared Error and correlation coefficient and were also obtained using cross-validation approach.

Additionally, the adaptive Soft Sensor is compared to the Locally Weighted Projection Regression (LPWR) which is an established adaptive modelling technique showing some similarities to the proposed algorithm. The parameters of the LWPR method were optimised for this modelling task and the optimal performance was achieved with following values: `'init_D'=4; 'init_alpha'=100; 'diag_only'=0; 'w_gen'=0.9; 'w_prune'=0.`

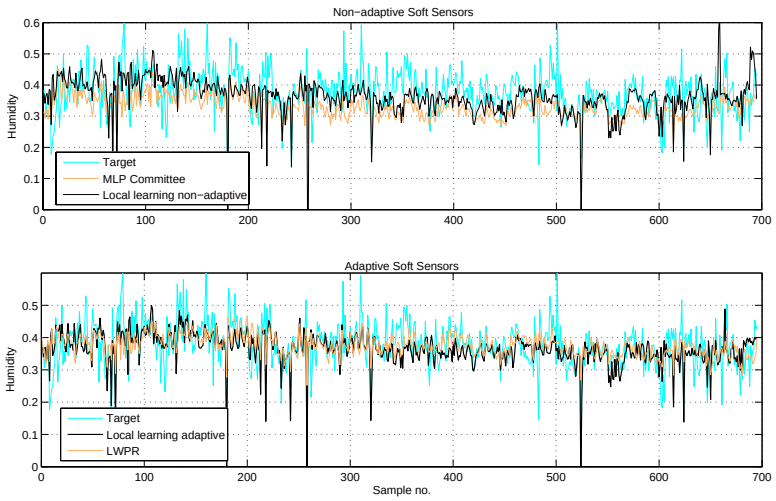
Experiment results: The local learning Soft Sensor partitions the historical, i.e. training, data into four receptive fields and builds four local experts. The results of the experiments are presented in Fig. 1 and in Tab. 1. In order to provide a fair comparison, Fig. 1(a) firstly presents the predictions of the non-adaptive models (MLP and non-adaptive local learning approach) and separately to this the adaptive Soft Sensor and the LWPR predictions. In the second part of Fig. 1(a), the adaptive version of the proposed Soft Sensor, as expected, further improves the performance and it follows the target values more accurately. Additionally it performs slightly better than the LWPR model.

The previous conclusions are more obvious in Fig. 1(b) which shows the predictions residuals of the four Soft Sensors after smoothing with a 30 samples long averaging filter. In ideal case the curves should get as close to the zero-line as possible. Comparing the two non-adaptive models, one can see that the local learning Soft Sensor (black-line) gets much closer to this goal than the MLP committee. The benefit of the adaptation technique proposed in this work can be also observed, i.e. the residuals of the local learning adaptive Soft Sensor are the closest to the zero-line.

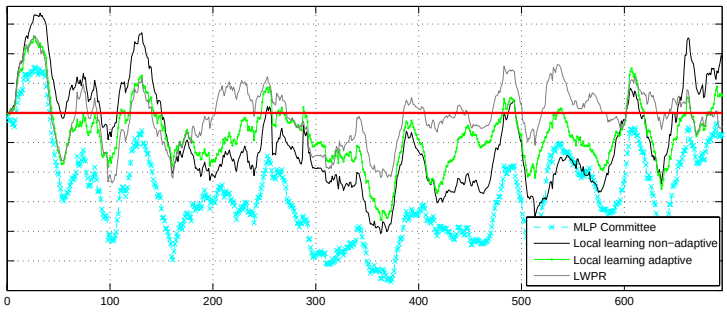
	Soft Sensor type	MSE	Corr. coeff.
non-adaptive	MLP Committee	0.0062	0.40
	Proposed local learning non-adaptive	0.0053	0.41
adaptive	LWPR	0.0042	0.48
	Proposed local learning adaptive	0.0040	0.50

Table 1. MSE and correlation coefficient performance of the three tested approaches

In summary Table 1 shows the Mean Squared Errors (MSE) and the correlation coefficients of the four Soft Sensors. The presented values confirm the previous results.



(a) Predictions



(b) Smoothen residuals

Fig. 1. Experimental results of the discussed four Soft Sensors

One can see a slight gain in MSE of the non-adaptive local learning model in comparison to the MLP which is achieved at lower model complexity at the same time. Further on the benefit of the two adaptive models can be observed. The best performance is achieved by the adaptive version of the proposed algorithm. Despite some similarities between the our algorithm and the LWPR, our model shows slightly better performance for this modelling problem.

4 Conclusions

The main contribution of this work is an adaptive Soft Sensor which is based on the local learning approach. Unlike the majority of so far presented approaches to soft sensing,

this algorithm also limits the effort which needs to be spent on the data pre-processing, model selection and Soft Sensor maintenance. The robustness of the Soft Sensor is achieved by deploying a set of local models. The partitioning of the input space further allows individual pre-processing and modelling of the receptive fields depending on the local complexity. The limitation of the maintenance requirements is achieved by applying a fully incremental adaptive approach. This approach is based on the modification of the combination weights during the run-time of the Soft Sensor and thus does not require the implementation of any model-specific adaptation technique. This factor further increases the flexibility of the model as any regression technique can be applied as local expert. The application possibilities of the proposed Soft Sensor are wide, it can be for example deployed to support the process control in form of an adaptive inferential control.

References

1. Fortuna, L.: Soft Sensors for Monitoring and Control of Industrial Processes. Springer
2. Jolliffe, I.T.: Principal Component Analysis. Springer
3. Abdi, H.: Partial least squares (pls) regression. Encyclopedia of Social Sciences, Research Methods. Thousand Oaks (CA): Sage (2003)
4. Dong, D., McAvoy, T.J., Chang, L.J.: Emission monitoring using multivariate soft sensors. In: American Control Conference, 1995. Proceedings of the. Volume 1.
5. Lin, B., Recke, B., Knudsen, J., Jrgensen, S.B.: A systematic approach for soft sensor development. Computers and Chemical Engineering **31**(5) (2007)
6. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, USA
7. Fortuna, L., Graziani, S., Xibilia, M.G.: Soft sensors for product quality monitoring in debutanizer distillation columns. Control Engineering Practice **13**(4) (2005)
8. Vapnik, V.N.: Statistical learning theory. Wiley New York
9. Jang, J.S.R., Sun, C.T., Mizutani, E.: Neuro-fuzzy and soft computing. Prentice Hall Upper Saddle River, NJ
10. Warne, K., Prasad, G., Siddique, N.H., Maguire, L.P.: Development of a hybrid pca-anfis measurement system for monitoring product quality in the coating industry. In: Systems, Man and Cybernetics, 2004 IEEE International Conference on. Volume 4.
11. Desai, K., Badhe, Y., Tambe, S.S., Kulkarni, B.D.: Soft-sensor development for fed-batch bioreactors using support vector regression. Biochemical Engineering Journal **27**(3) (2006)
12. Jacobs, R.: Adaptive mixtures of local experts. Neural Computation **3**(1) (1991)
13. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally weighted learning. Artificial Intelligence Review **11**(1) (1997)
14. Geman, S., Bienenstock, E., Doursat, R.: Neural networks and the bias/variance dilemma. Neural Computation **4**(1) (1992)
15. French, R.: Catastrophic forgetting in connectionist networks: Causes, consequences and solutions. Trends in Cognitive Sciences **3**(4) (1999)
16. Vijayakumar, S., D'Souza, A., Schaal, S.: Incremental online learning in high dimensions. Neural Computation **17**(12) (2005)
17. Parzen, E.: On estimation of a probability density function and mode. Annals of Mathematical Statistics **33**(3) (1962)
18. Gosset, W.S.: The probable error of a mean. Biometrika **6**(1) (1908)