A Neuro-fuzzy Approach for Sensor Network Data Cleaning

Alfredo Petrosino and Antonino Staiano

Dipartimento di Scienze Applicate, Università di Napoli "Parthenope",
Via A. De Gasperi, 5 I-80133 Napoli, Italy

Abstract. Sensor networks have become an important source of data with numerous applications in monitoring various real-life phenomena as well as industrial applications and traffic control. However, sensor data are subject to several sources of errors as the data captured from the physical world through these sensor devices tend to be incomplete, noisy, and unreliable, thus yielding imprecise or even incorrect and misleading answers which can be very significative if they result in immediate critical decisions or activation of actuators. Traditional data cleaning techniques cannot be applied in this context as they do not take into account the strong spatial and temporal correlations typically present in sensor data, so machine learning techniques could greatly be of aid. In this paper, we propose a neuro-fuzzy regression approach to clean sensor network data: the well known ANFIS model is employed for reducing the uncertainty associated with the data thus obtaining a more accurate estimate of sensor readings. The obtained cleaning results show good ANFIS performance compared to other common used model such as kernel methods, and we demonstrate its effectiveness if the cleaning model has to be implemented at sensor level rather than at base-station level.

Keywords: Sensor networs, data cleaning, regression, neuro-fuzzy.

1 Introduction

Sensor networks are deployed in various domains to acquire information about different physical phenomena in real-time. Each sensor node has embedded processor capability, and potentially has multiple onboard sensors, operating in the acoustic, seismic, infrared (IR), and magnetic modes, as well as imagers and microradars. The data acquired are typically not directly usable as they suffers from three problems: a) noise because of inaccuracy in hardware sensing and transmission and unfavorable environmental conditions and limited battery power further exacerbates this problem; b) missing values usually occur due to packet loss and node failure; c) incompleteness, since sensors sample continuous physical phenomena at discrete time intervals. All these problems seriously impact the quality of data obtained from such networks. The aim of the industry, indeed, is to manufacture tiny, cheap sensors that can be deployed everywhere and disposed when depleted. Consequently, noise, imprecision and inaccuracies

B. Apolloni et al. (Eds.): KES 2007/ WIRN 2007, Part III, LNAI 4694, pp. 140–147, 2007.

are inevitable in these cheap sensors [3]. It is extremely important that data from these sensors be reliable since actions are usually taken based on their readings. Dirty data can lead to detrimental effects since they may be used in critical decisions or in the activation of actuators. This is why data cleaning has assumed such a prominent role in last years in sensor networks and calls for new automated approach to address its goal. Data cleaning is a procedure inherited from data mining [4] where the focus has been primarily in the context of information integration and data-warehousing [11],[12]. However, the nature of sensor network data is inherently different, and previous approaches cannot always be applied directly in this domain. The work done on data cleaning in the context of sensor network data is still limited but the most use statistical and machine learning techniques, such as Bayesian approach [3], Kalman Filtering [10] and Kernel methods [7] just to mention a few. According to these approaches, the aim is to construct a regression model and, through which, reduce the noise in the sensor readings and fill in the missing fields.

The present paper is in the same direction. We propose to adopt a neuro-fuzzy system, namely, ANFIS [5], that provides us a way to deal with imprecision and vagueness, typical of sensor data, thanks to fuzzy sets and fuzzy reasoning [9] while giving us the learning ability of neural networks [1]. Furthermore, since cleaning can be performed either at the individual sensors or at the base station [3], ANFIS turns to be a lightweight model if the sensor level implementation is chosen.

The paper is organized as follows. Section 2 reports the proposed method, whilst in section 3 we briefly overview the ANFIS theoretical foundation and provide some details on kernel methods we used as benchmark statistical learning model for regression. In section 4 we describe the application of the ANFIS model to the Intel Berkeley Lab Data set[8], illustrating the obtained results and comparing them with kernel methods. Finally, section 5 provides some concluding remarks.

2 Regression Based Sensor Network Data Cleaning

The sensor network data cleaning procedure we propose behaves as follows. The idea is firstly to build a regression model which is used to approximate the point distribution coming from the sensor network readings. In order to compute the regression model for the sensors the learning algorithms need to be trained on a past time interval. Training is accomplished off-line, mainly at the base-station level. Once the regression model is derived from a training sample of sensor network readings, it describes the behavioral model of the sensors. This behavior is adopted to correct the sensor readings in two ways:

- replacing the readings of the training sample by the regression function itself which can be also used to fill in the missing readings;
- by correcting new sensor readings on the base of the regression model predictions. This step could be performed either at base-station level and at sensor level. The latter is done on-line.

Basically, since we are modeling the behavior of each sensor with respect to the entire sensor network and the space-temporal relations between them, the error corresponding to each sensor is a random variable whose error may be approximated by the regression model RMSE (Root Mean Square Error) estimation on the training sample. Determining the behavioral model of each sensor in the network corresponds to derive the sensor error models. Therefore, if the difference between new sensor readings (y_{id}) and the model estimates ($\hat{y}_{id}$) are less than the RMSE, then new sensor readings are considered reliable, otherwise the new readings are replaced by the model estimates plus (or minus) the RMSE. In this way the new readings fall between $[\hat{y} - RMSE, \hat{y} + RMSE]$.

Formally speaking, let y_{id}, $\hat{y}_{id}$ and yc_{id} be respectively the new sensor (with identifier number id) reading, the sensor reading estimate and the corrected sensor reading, and $\hat{e}$ the RMSE, we define:

$$yc_{id} = \begin{cases} y_{id} & \text{if } |y_{id} - \hat{y}_{id}| < \hat{e} \\ \hat{y}_{id} \pm \hat{e} & \text{if } |y_{id} - \hat{y}_{id}| \geq \hat{e}. \end{cases} \quad (1)$$

In this way, modeling the temperature behavior as example, from equation 1, if temperature values suddenly increase for an accidental reason, the reading will be suitably corrected without altering the peculiar characteristics of the observed physical phenomena (the temperature values in a given environment).

To deal with the uncertainty of data readings, we chose to model it by adopting fuzzy memberships and fuzzy rules in the modeling phase. Specifically, we adopt a neural network based fuzzy logic system (NFS) for modeling. NFSs merge ideas from fuzzy control and neural networks, and possess the advantages of both neural networks (e.g. learning abilities, optimization abilities) and fuzzy control systems (e.g. human like IF-THEN rule thinking and ease of incorporating expert knowledge). The main purpose is to apply neural learning techniques to find and tune the parameters and/or structure of the system. Two major types of tuning are required: structural tuning and parametric tuning. Structural tuning involves tuning of the structure of fuzzy logic rules such as the number of variables to account for, the number of rules and the conjunctions that constitute them, and so on. Once a satisfactory structure of the rules is obtained, the NFS needs to perform parametric tuning. In this parameter-learning phase, the possible parameters to be tuned include these associated with membership functions such as centers, widths, and slopes, the parameters of the parameterized connectives and the weights of the fuzzy logic rules. Supervised learning, which require a teacher to specify the desired output vector is suitable for parameter learning to adjust the parameters of fuzzy logic rules and/or membership functions for the desired output in neuro-fuzzy control systems. In cases where the membership functions are differentiable, gradient-based learning methods (e.g. Back-Propagation algorithm) for parameter learning can easily derived. In the next section we briefly describe the neuro-fuzzy system we adopted to build the regression model and provide a brief insight into kernel methods for regression which are used for comparative purposes.

3 Neuro-fuzzy and Kernel Models for Regression

3.1 ANFIS

Among NFSs, an important role can be played by Adaptive Neuro-Fuzzy Inference Systems (ANFIS)[13]. In fact, such computational models can be used as predictors by means of a suitable transformation of the prediction problem into a function approximation one. An ANFIS network performs the approximation of an unknown mapping $y = f(\mathbf{x})$, $f : R^N \rightarrow R$, by implementing a fuzzy inference system constituted by M rules of Sugeno first-order type. The $k-th$ rule, $k = 1, \ldots, M$ gets the form:

$$\texttt{If } x_1 \texttt{ is } B_1^{(k)}, \ldots, \texttt{ and } x_N \texttt{ is } B_N^{(N)} \texttt{ then } y^{(k)} = \sum_{j=1}^{N} p_j^{(k)} x_j + p_0^{(k)}, \quad (2)$$

where $\mathbf{x} = (x_1, \ldots, x_N)$ is the input pattern and $y^{(k)}$ is the output associated to the rule. The antecedent part of the rule is characterized by the membership functions $\mu_{B_j^{(k)}}(x_j)$ of the fuzzy input variables $B_j^{(k)}, j = 1, \ldots, N$, while the consequent part is characterized by the coefficients $p_j^{(k)}, j = 0, \ldots, N$ of the crisp output $y^{(k)}$. Several alternatives are possible for choosing the fuzzification type of crisp inputs , the composition of input membership functions, and the way rule outputs are combined [5]. By using the commonly adopted options in this regard, the overall ANFIS output will be obtained by the following approximation model:

$$\overline{y} = \frac{\sum_{k=1}^{M} \mu_{\overline{B}^{(k)}}(\mathbf{x}) y^{(k)}}{\mu_{\overline{B}^{(k)}}(\mathbf{x})}, \quad (3)$$

where $\overline{y}$ is the estimated output of the actual value $y = f(\mathbf{x})$, and $\mu_{\overline{B}^{(k)}}(\mathbf{x})$ is the overal input membership function of the $k-th$ rule, which can be obtained either by a direct estimation procedure or by the composition of the corresponding input membership functions, i.e. $\mu_{\overline{B}_j^{(k)}}(x_j), j = 1 \ldots, N$. For further details, the reader main refer to [5].

3.2 Kernel Methods

Kernel methods [6] is a class of pattern recognition technique, in which the training data points, or a subset of them, are kept and used also during the prediction phase. The kernel approach to pattern analysis first embeds the data in a suitable feature space, and then uses algorithms based on linear algebra, geometry and statistics to discover patterns in the embedded data. For models which are based on a fixed nonlinear feature space mapping $\phi(\mathbf{x})$, the kernel function is given by the relation

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}'). \quad (4)$$

Let us consider a linear regression model whose parameters are determined by minimizing a regularized sum-of-squares error function given by

$$J(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left\{\mathbf{w}^T\phi(\mathbf{x}_n - t_n)\right\}^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \tag{5}$$

where $\lambda \geq 0$. By solving with respect to $\mathbf{w}$ and applying some matrix algebra manipulations (see [1] for details) we obtain the following prediction for a new input $\mathbf{x}$

$$y(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) = \mathbf{a}^T\mathbf{\Phi}\phi(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T(\mathbf{K} + \lambda\mathbf{I}_N)^{-1}\mathbf{t} \tag{6}$$

where we have defined the vector $\mathbf{k}(\mathbf{x})$ with elements $k_n(\mathbf{x}) = k(\mathbf{x}_n, \mathbf{x})$. Thus we see that the dual formulation allows the solution to the least-squares problem to be expressed entirely in terms of the kernel function $k(\mathbf{x}, \mathbf{x}^{'})$. $\mathbf{K}$ is the Gram matrix and $\mathbf{t}$ is the target vector corresponding to the training data. The advantage of the dual formulation is that it is expressed entirely in terms of the kernel function $k(\mathbf{x}, \mathbf{x}^{'})$. We can therefore work directly in terms of kernels and avoid the explicit introduction of the feature vector $\phi(\mathbf{x})$, which allows us implicitly to use feature spaces of high, even infinite, dimensionality.

4 Experimental Results

Our proposed approach using ANFIS, has been evaluated on the Intel Lab data set [8] of public domain, containing readings collected from 54 sensors deployed in the Intel Berkeley Research Lab. Sensors with weather boards collected humidity, temperature, light and voltage values once every 31 seconds. The complete data set consists of approximately 2.3 million readings collected from these sensors. The format of the data set is as follows: date, time, epoch, mote ID, temperature, humidity, light, and voltage. Here we considered time, temperature, humidity, voltage and light as input features leaving, in turn, one feature out from this set (except time) as output variable to model

We conceived two groups of experiments, namely: (a) single sensor modeling, according to which we model the behavioral model of every sensor of the entire network, i.e a model for every sensor out of 54 sensors; (b) near group sensors, in which we model the behavior of a chosen sensor by also considering the readings of some (typically four) neighbor sensors, all done for every sensor out of 54 sensors. The experiments have the aim to prove that, trained in this way, ANFIS is able to hold the temporal and also the spatial relations between sensors. The training and test sets were built by sampling the sensor reading each hour, covering one calendar month (February-March). So doing, each training set correspond to 960 sensor readings (about three weeks of time period) and each test set to 180 readings (about one week). ANFIS needs to model the time series coming from our data sets. For comparative aim, we also considered the results obtained by a polynomial regression kernel (the polynomial kernel was chosen after a group of experiments aimed to assess the most performing kernel with these data sets).

In the following, we only illustrate, due to page limits, the obtained ANFIS results on sensor 1 and sensor 1 and its four neighbors when modeling the temperature. The results are then compared with those obtained by adopting the polynomial kernel. Fig. 1, shows the regression curves approximating the sensor 1 temperature behavioral model, both for ANFIS and polynomial kernel. The quality of the approximations on the test set are comparable as well as the RMSE obtained on the same set (6.21 for polynomial kernel and 8.21 for ANFIS). The temperature models for sensor 1, on the basis of its four neighbors are depicted in Fig. 2. Here, the models are qualitatively worst especially for ANFIS, even though the overall behavior of the sensor is well reproduced without following the sudden temperature range which occurs in the last tail of the readings. The obtained RMSE confirm this; we got 9.86 for polynomial kernel and 12.81 for ANFIS. The second experiment on the sensor 1 and its neighbors, clearly prove that the ANFIS model is inherent able to consider the spatial relationship between sensors.

Next, we applied to sensor 1 and sensor 1 and its neighbors, through the behavioral model obtained by ANFIS, the correction (1) to the sensor readings, obtaining the cleaned values which give rise to the curves showed in Fig. 3. The considered RMSE for the cleaning is that one obtained on the training sets (1.56 for sensor 1 and 1.88 for sensor 1 and its neighbors). As emerges from the figures, the cleaning procedure makes possible to hold the normal behavior of the sensors without acquiring outlier values due to exceptional events or to noise.

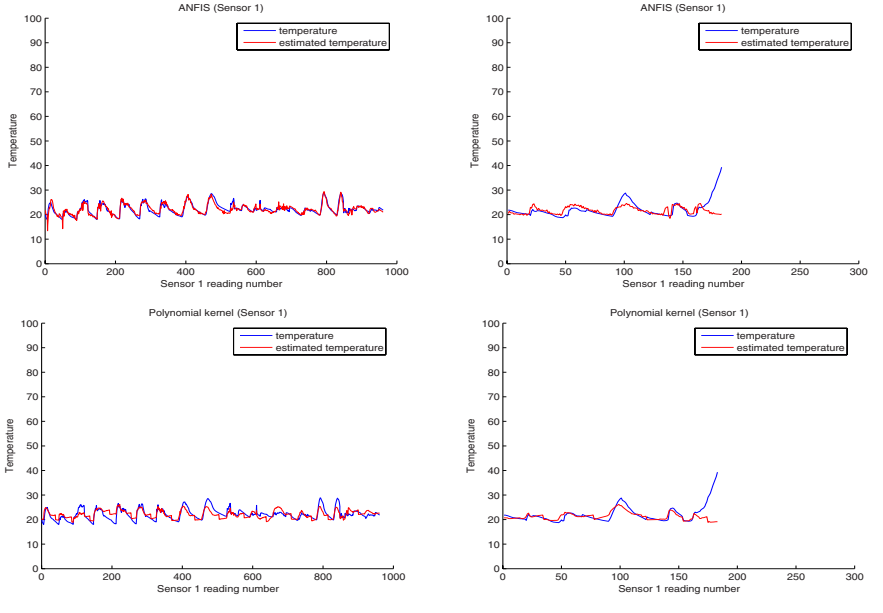

Fig. 1. Sensor 1 temperature estimate. Left: Training sets, Right: Test sets. Up: The model by ANFIS, Down: The model by polynomial kernel.

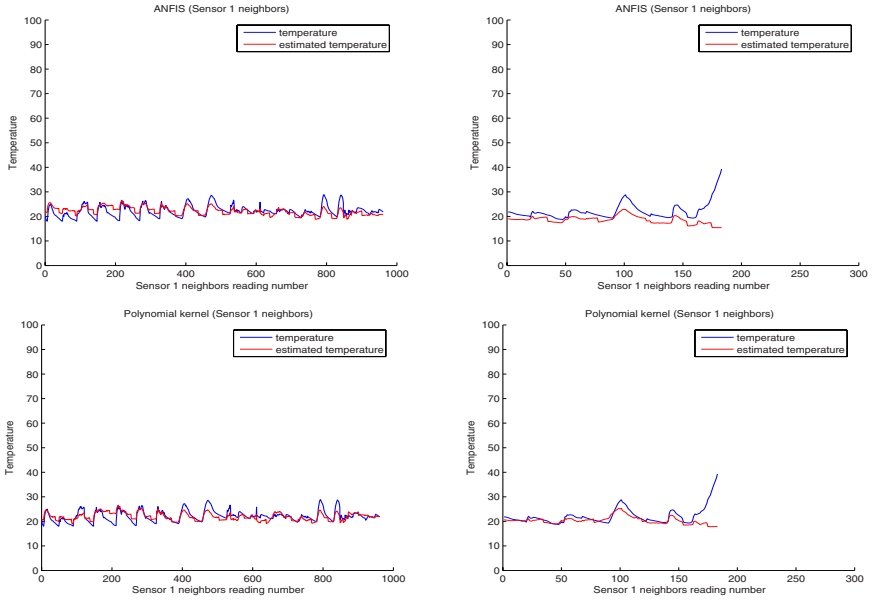

Fig. 2. Sensor 1 and neighbors temperature estimate. Left: Training sets, Right: Test sets. Up: The model by ANFIS, Down: The model by polynomial kernel.

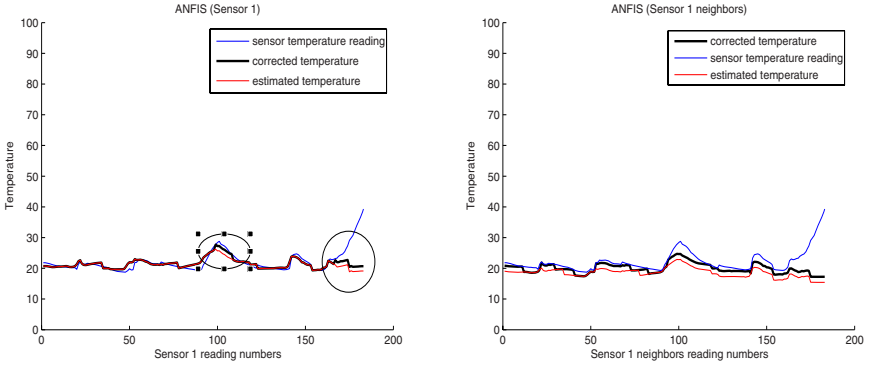

Fig. 3. Temperature reading corrections: Left: Sensor 1; Right: Sensor 1 and its four neighbors. The circled areas highlight the points where the difference between model estimates and sensor readings exceed the RMSE.

5 Conclusions

We proposed a neuro-fuzzy approach to sensor network data cleaning. Basically, the cleaning procedure is based on the computation of the behavioral model of the sensors through a regression approximation obtained by ANFIS and correcting the sensor new reading on the base of the RMSE provided by ANFIS

on the training set. ANFIS was compared with a polynomial regression kernel. This latter model is able to provide more accurate results, however, the ANFIS approach is preferable since its computational lightweight. This is because, the kernel methods are memory based learning systems, i.e. systems which need the training data set to compute the prediction on the test sets. This makes kernel methods completely inadequate for a sensor level implementation. ANFIS, on the other hand, while giving only little bit less accurate predictions, is more suitable for both base-station level and sensor level implementations.

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
2. Bychkovskiy, V., Megerian, S., Estrin, D., Potkonjak, M.A.: A Collaborative Approach to in-place Sensor Calibration. In: Zhao, F., Guibas, L.J. (eds.) IPSN 2003. LNCS, vol. 2634, Springer, Heidelberg (2003)
3. Elnahwary, E., Nath, B.: Cleaning and Querying Noisy Sensors. IN WSNA'03. In: Proceedings of the 2nd ACM International Conference on Wireless Sensors and Applications, pp. 78–87. ACM Press, New York (2003)
4. Hand, D.J., Mannila, H., Smyth, P.: Principles of Data Mining. MIT Press, Cambridge (2001)
5. Nauck, D., Kruse, R.: Neuro-Fuzzy Models in Fuzzy Rule Generation. In: Bezdek, J.C., Dubois, D., Prade, H. (eds.) Fuzzy Sets in Approximate Reasoning and Information Systems, Kluwer, Dordrecht (1999)
6. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)
7. Tan, Y.L., Sehgal, V., Shahri, H.H.: SensoClean: Handling Noisy and Incomplete Data in Sensor Networks using Modeling. Technical Report, University of Maryland (2005)
8. Intel Berkeley Laboratory Data, `http://berkeley.intel-research.net/labdata/`
9. Pedrycz, W., Kandel, A., Zhang, Y.Q.: Neurofuzzy Systems. In: Nguyen, H.T., Sugeno, M. (eds.) Fuzzy Systems Modeling and Control. The Handbook on Fuzzy Sets, Kluwer Academic Publishers, Dordrecht (1998)
10. Spanos, D.P., Olfati-Saber, R., Murray, R.M.: Approximate Distributed Kalman Filtering in Sensor Networks with Quantifiable Performance. In: Fourth International Symposium on Information Processing in Sensor Networks, pp. 133–139 (2005)
11. Galhardas, H., et al.: Declarative data cleaning: Language, model, and algorithms. In: VLDB, pp. 371–380 (2001)
12. Raman, V., Hellerstein, J.M.: Potter's Wheel: An Interactive Data Cleaning System. The VLDB Journal, 381–390 (2001)
13. Jang, R., ANFIS,: Adaptive-Network-based Fuzzy Inference Systems. IEEE Trans. on Systems, Man, and Cybernetics 23, 665–685 (1993)