An adaptive modular approach to the mining of sensor network data.

Gianluca Bontempi, Yann-Aël Le Borgne*
ULB Machine Learning Group
Université Libre de Bruxelles, Belgium
email: {gbonte,yleborgn}@ulb.ac.be

Abstract

This paper proposes a two-layer modular architecture to adaptively perform data mining tasks in large sensor networks. The architecture consists in a lower layer which performs data aggregation in a modular fashion and in an upper layer which employs an adaptive local learning technique to extract a prediction model from the aggregated information. The rationale of the approach is that a modular aggregation of sensor data can serve jointly two purposes: first, the organization of sensors in clusters, then reducing the communication effort, second, the dimensionality reduction of the data mining task, then improving the accuracy of the sensing task.

1 Introduction.

There are plenty of potential applications for intelligent sensor networks: distributed information gathering and processing, monitoring, supervision of hazardous environments, intrusion detection, cooperative sensing, tracking.

The ever-increasing use of sensing units asks for the development of specific data mining architectures. What is expected from these architectures is not only accurate modeling of high dimensional streams of data but also a minimization of the communication and computational effort demanded to each single sensor unit.

The simplest approach to the analysis of sensor network data makes use of a centralized architecture where a central server maintains a database of readings from all the sensors. The whole analysis effort is localized in the server, whose mission is to extract from the flow of data the high-level information expected to be returned by the monitoring system. If we assume that reasonable-size sensor networks will be made of thousands of nodes, the limitation of this approach is strikingly evident: the number of messages sent in the system as well as the number of variables of the data mining task are too large to be managed efficiently.

It has been suggested in literature that alternative architectures are to be preferred in applications where neighboring sensors are likely to have correlated readings [6]. This is the case of *aggregating systems* which, according to the definition of [10], are systems where the data obtained from the different source nodes can be aggregated before being transmitted along the network. In these systems, we can imagine the existence of intermediary nodes (*aggregators*) having the capability to fuse the information from different sources. Sensor networks for weather forecasting and monitoring are examples of aggregating systems. The authors of [6] showed that a compression of the sensor information can be performed at local level then reducing the amount of communication and the bandwidth required for the functioning of the system.

At the same time techniques of data compression, like Principal Component analysis (PCA) or Independent Component Analysis (ICA) [8], are often used in data mining to reduce the complexity of modeling tasks with a very large number of variables. It is well known in the data mining literature that methods for reducing complexity are beneficial for several reasons: improvement of the accuracy and intelligibility of the model, reduced storage and time requirements.

The rationale of the paper is that a modular organization of the sensor network can be used to jointly address the two main issues in mining sensor network data: the minimization of the communication effort and the accurate extraction of high-level information from massive and streaming datasets.

In particular this paper proposes a data driven procedure to configure a two-layer topology of a sensor network (Figure 1) made of

1. a lower level whose task is to organize the sensors in clusters, compress their signals and transmit the aggregate information to the upper level,
2. an upper level playing the role of a data mining server which uses the aggregate information to

*Supported by the **COMP²SYS** project, sponsored by the HRM program of the European Community (MEST-CT-2004-505079)

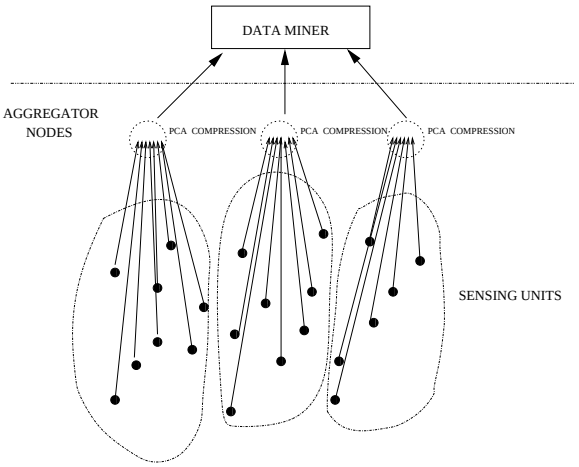

Figure 1: The black dots represent the sensing units. The dotted circles represent the aggregator nodes which carry out the fusion of data coming from neighboring sensors before sending the aggregated signals up to the data mining server.

carry out the required sensing task.

We focus here on problems where the sensors are used to perform a supervised learning (e.g. classification, regression or prediction) task: examples could be the classification of traffic fluidity on the basis of route sensing units or the prediction of a wave intensity on the basis of sensors scattered in the ocean. Our approach consists in using a historical data set to find the best way to combine the measures of neighboring sensors such that the accuracy of the prediction model based on such aggregate measures is optimized. The design procedure relies on an iterative optimization procedure which loops over five steps: (i) a partition of the sensing units in proximity clusters, (ii) the compression of the signals of each cluster of sensors, (iii) the aggregation and transmission of the compressed signals to the upper data mining server, (iv) the training of the prediction model in the data mining server, and (v) the assessment of the partition according to multiple criteria, like the prediction accuracy of the data mining model and the energy and transmission requirements of the resulting network. The best partition which is returned by this multi-criteria optimization procedure can be used as a template for the topological organization of sensors.

An important issue in mining sensor network data concerns the streaming and possibly non stationary nature of data. Non stationarity may be due to changes in the phenomenon underlying the measures as well to sensor malfunctioning and/or modifications of their geographical location. In order to address this aspect we have recourse to adaptive features at both levels of our architecture. At the lower sensor integration level we use an effective sequential implementation of the Principal Component Analysis (PCA) technique: the PAST algorithm [11]. The upper data mining module uses an adaptive lazy learning (LL) technique [1] characterized by a fast training phase and an effective treatment of non stationarity.

The experimental section of the paper presents some preliminary results obtained by adopting the proposed two-layer architecture in the context of a simulated monitoring task: measuring the wavelength of a two dimensional wave in situation of scattering.

2 The problem

Consider a sensor network $\mathcal{S}$ made of S sensors where P is a $[S, 3]$ matrix containing the three-dimensional coordinates of the S sensors and

$$x(t) = \{s_1(t), s_2(t), \ldots, s_S(t)\} \tag{2.1}$$

is the state (or snapshot) of the sensor network at time t. Suppose we intend to employ $\mathcal{S}$ to perform a supervised learning task, for example a regression problem

$$y(t) = f(x(t)) + \varepsilon(t) \tag{2.2}$$

where y is the variable to be predicted at time t on the basis of the state $x(t)$ of the network $\mathcal{S}$ and ε is usually thought as the term including modeling error, disturbances and noise.

If we have available a finite dataset $D_N = \{\langle x(t_i), y(t_i)\rangle, i = 1, \ldots, N\}$ of N input-output observations, this problem can be tackled as a conventional regression problem, by first estimating an approximator of f on the basis of D_N and then using this estimator as a predictor of y.

However, if, like in the case of sensor networks, the number S is huge, the mapping f is non-stationary and the data are collected sequentially, conventional techniques reach rapidly their limits. In particular, the large dimensionality of the problem asks for feature selection problem as well as the streaming aspect of the problem requires sequential (also called recursive) estimation approaches.

This paper proposes an approach to the problem of data mining in sensor networks which tries to conciliate the needs for an accurate prediction of the output y with the constraints related to energy reserves, communication bandwidth and sensor computational power.

The following subsections will rapidly sketch the two computational modules used in our approach: the recursive PCA and the adaptive Lazy Learning. Section 5 will describe how these modules are combined in our architecture for mining sensor networks.

3 PCA compression techniques

As discussed above, each data mining problem in the context of sensor network data with large S has to face the problem of reducing dimensionality. Existing techniques for feature selection (for an up-to-date state of the art on feature selection see [7]) can be grouped into two main approaches: the wrapper and the filter approach. In the wrapper approach [9] the feature subset selection algorithm exists as a wrapper around the learning algorithm, which is often considered as a black box able to return (e.g. via cross-validation) an evaluation of the quality of a feature subset. On the contrary, the filter approach selects features using a preprocessing step independently of the learning algorithm.

In this paper we will adopt the Principal Component analysis (PCA) technique, an instance of the filter approach. PCA is a classic technique in statistical data analysis, feature extraction and data compression [8]. Given a set of multivariate measurements, its goal is to find a smaller set of variables with less redundancy, that would give as good a representation as possible. In PCA the redundancy is measured by computing linear correlations between variables. PCA entails transforming the n original variables $x_1, \ldots, x_n$ into m new variables $z_1, \ldots, z_m$ (called *principal components*) such that the new variables are uncorrelated with each other and account for decreasing portions of the variance of the original variables. Consider a vector x of size n and a matrix X containing N measures of the vector x. The m principal components

$$z_k = \sum_{j=1}^{n} w_{jk} x_j = w_k^T x, \qquad k = 1, \ldots, m \tag{3.3}$$

are defined as weighted sums of the elements of x with maximal variance, under the constraints that the weights are normalized and the principal components are uncorrelated with each other. It is well-known from basic linear algebra that the solution to the PCA problem is given in terms of the unit-length eigenvectors $e_1, \ldots, e_n$ of the correlation matrix of x. Once ordered the eigenvectors such that the corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$ satisfy $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$, the principal component z_k is given by $z_k = e_k^T x$. It can be shown that the PCA problem can be also put in the form of a minimum mean-square error compression of x. This means that the computation of the w_k for the first m principal components is equivalent to find the orthonormal basis $w_1, \ldots, w_m$ that minimizes

$$J_{PCA} = \frac{1}{N} \sum_{t=1}^{N} \|x(t) - \sum_{k=1}^{m} (w_k^T x(t)) w_k\|^2 \tag{3.4}$$

If we denote $W = (w_1, \ldots, w_m)^T$ where W is a matrix of size $[m, n]$ we have

$$J_{PCA} = \frac{1}{N} \sum_{t=1}^{N} \|x(t) - W^T W x(t)\|^2 \tag{3.5}$$

What is appealing in this formulation is that a recursive formulation of this least-squares problem is provided by the Projection Approximation Subspace Tracking (PAST) algorithm proposed by [11]. This algorithm, based on the recursive formulation of the least squares problem, has low computational cost and makes possible an updating of the principal components as new observations become available.

Once the matrix W is computed a reduction of the input dimensionality is obtained by transforming the input matrix X into the matrix $Z = XW^T$ and by transforming the regression problem of dimensionality n into a problem of dimensionality m in the space of principal components.

At this step the question arises of how to choose m. The techniques more commonly used rely either on the absolute values of the eigenvalues or on procedures of cross-validation [8].

4 The Lazy Learning algorithm

In supervised learning literature a possible way to classify learning techniques relies on the dichotomy: local memory-based versus global methods. Global modeling builds a single functional model of the dataset. This has traditionally been the approach taken in neural networks [2] and other form of non-linear statistical regression. The available dataset is used by a learning algorithm to produce a model of the mapping and then the dataset is discarded and only the model is kept. Local algorithms defer processing of the dataset until they receive request for information (e.g. prediction or local modeling). The classical nearest neighbor method is the original approach to local modeling. A database of observed input-output data is always kept and the estimate for a new operating point is derived from an interpolation based on a neighborhood of the query point.

The data mining architecture proposed in this paper adopts the Lazy Learning (LL) algorithm proposed by [1], an instance of the local modeling approach that, on a query-by-query basis, tunes the number of neighbors using a local cross-validation criterion. For a detailed description of the approach see also [5] et references therein. The LL algorithm, publicly available in a MATLAB and R implementation[1], proved to be successful in many problems of non-linear data analysis and time series prediction [5, 3, 4].

[1] http://iridia.ulb.ac.be/∼lazy

This paper illustrates and validates the use of Lazy learning for the task of mining sensor network data. The author deems that this algorithm presents a set of specific features which makes of it a promising tool in the sensor network context:

The reduced number of assumptions. Lazy Learning assumes no a priori knowledge on the process underlying the data. For example, it makes no assumption on the existence of a global function describing the data and no assumptions on the properties of the noise. The only available information is represented by a finite set of input/output observations. This feature is particularly relevant in real datasets where problems of missing features, non stationarity and measurement errors make appealing a data-driven and assumption-free approach.

On-line learning capability. The Lazy Learning method can easily deal with on-line learning tasks where the number of training samples increases or the set of input variables changes with time. In these cases, the adaptiveness of the method is obtained by simply adding new points to the stored dataset or restricting the analysis to the accessible inputs. This property makes the technique particularly suitable for monitoring problems where the number of samples increases with time or the set of available signals may vary due to sensor malfunctioning and/or bad communications.

Modeling non-stationarity. LL can deal with time-varying configurations where the stochastic process underlying the data is non-stationary. In this case, it is sufficient to interpret the notion of neighborhood not in a spatial way but both in a spatial and temporal sense. For each query point, the neighbors are no more the samples that have similar inputs but the ones that both have similar inputs and have been collected recently in time. Therefore, the time variable becomes a further precious feature to consider for accurate prediction.

5 A two-layer architecture for mining sensor network data

The conventional techniques of dimensionality reduction by PCA described in Section 3 aim to reduce the collinearity or linear relationships existing between the different inputs. This technique creates new variables obtained by combining linearly the original inputs. Typically, this linear combination assigns a weight different from zero to all the original inputs.

Consider now the idea of applying the PCA to a problem of regression where data are generated by a sensor network $\mathcal{S}$ with a large number of sensing units S. Although the PCA allows a reduction of the input space from S to m and possibly an improvement of the prediction accuracy, the resulting prediction model needs the values of all the S sensing units in order to perform its computation. This requires inevitably a centralized architecture where all the sensors are required to transmit their measures to the central server.

A distributed architecture cannot take advantage of this way of performing dimensionality reduction. Our approach consists instead in applying the PCA technique not to the whole of sensors but in a modular fashion to subsets made of neighboring sensing units.

Suppose that a partition $\mathcal{P}$ of the S sensors into C clusters of neighboring sensing units is available. Let n_c, $c = 1, \ldots, C$, be the number of sensing units contained into the cth cluster.

The algorithm we propose consists into applying C separate recursive PAST computations to each of the cluster. Let m_c be the number of principal components which are retained for each cluster and z_c the $[m_c, 1]$ vector of transformed variables returned by the PCA applied to the cth cluster.

The original supervised learning problem (2.2) whose input space has dimensionality S is now replaced by a new supervised learning problem featuring an input space $[z_1, \ldots, z_C]$ of dimensionality $\sum_{c=1}^{C} m_c$.

A cross-validation procedure can be adopted to assess the quality of the prediction model trained in the transformed space.

This assessment figure can be used as a measure of quality of the repartition $\mathcal{P}$. It is then possible to implement an iterative procedure that explores the space of possible repartitions aiming to find the one with the best prediction accuracy. This optimization procedure could be turned into a multi-criteria optimization by taking into account together with the prediction accuracy also a measure of the communication cost of the repartition under analysis.

Note that the outcome of the resulting procedure is a repartition of the nodes of the sensor network into C clusters, a procedure for aggregating locally the measures of the n_c nodes composing the cth cluster and a model at the server level which returns the high-level prediction.

6 Experimental results

The two-layer adaptive architecture has been tested on a simulated dataset generated by solving numerically the Helmholtz equation, an elliptic partial differential

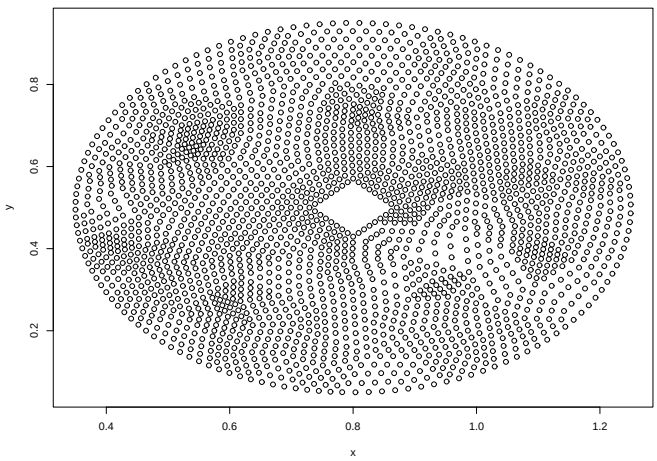

Figure 2: The distribution of the sensing units

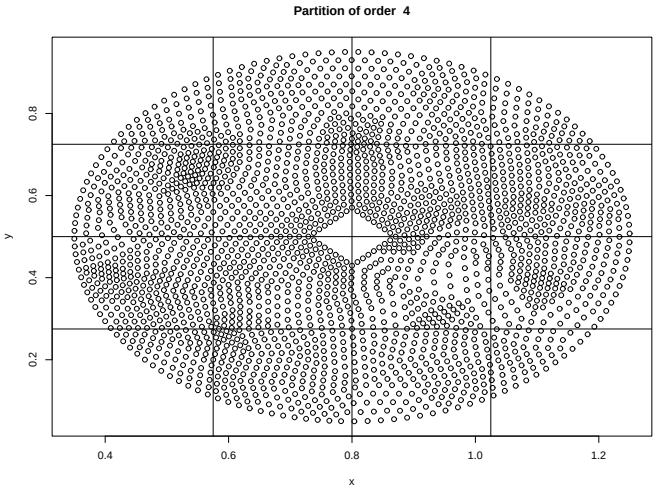

Figure 3: The uniform partition $\mathcal{P}^{(4)}$ of the sensor network in $4^2 = 16$ clusters.

equation,

$$\nabla^2 u + k^2 u = 0 \tag{6.6}$$

where $u(P_x, P_y, t) = s(t)$ is the value of scalar field in the point $[P_x, P_y]$ at time t and k is the wave number.

The Helmholtz equation governs some important physical phenomena, including the potential in time harmonic acoustic and electromagnetic fields. In our example the equation is used to model the waves reflected from a square object in a homogeneous medium. The wave number is related to the wavelength λ by the relation $k = \frac{2\pi}{\lambda}$.

We perform several simulations with 30 different wave numbers ranging from 1 to 146. For each wave number we collect the value of the u field in 50 time instants. Altogether, we collected $N = 1500$ measures of the wave field in a mesh of $S = 2732$ points depicted in Figure 2.

We formulate a regression problem where the goal is to predict the wave number k on the basis of the S measurements at time t returned by the simulated sensor network. The regression is performed on the output $y = k + \varepsilon$ that is a corrupted version of the signal k obtained by adding to k a random gaussian additive noise with standard deviation $\sigma_\varepsilon = \sigma_k/3$.

We consider a sequence of partitions $\{\mathcal{P}^{(1)}, \mathcal{P}^{(2)}, \ldots, \}$ where $\mathcal{P}^{(d)}$ is an uniform lattice with $d-1$ divisions along each dimension. This means that the partition $\mathcal{P}^{(d)}$ decomposes the sensor field into $C = d^2$ clusters. For illustration, the partition $\mathcal{P}^{(4)}$ is reported in Figure 3. Note that the partition $\mathcal{P}^{(1)}$ is equivalent to a centralized configuration where all sensors transmit their unprocessed measures to the central server.

Given a partition, a recursive PCA is performed on each cluster to return the first $m_c = 2$ principal components. Then the resulting aggregated signals are transmitted to the data mining server which performs the regression modeling by using the adaptive Lazy Learning algorithm described in Section 4. The quality of the data mining step is assessed by a ten-fold cross validation strategy. At each run of the cross-validation, we use $N_{tr} = 1350$ samples for the training set and the remaining $N_{ts} = 150$ samples for the test set. The prediction accuracy is computed by averaging the Normalized Mean Square Error (NMSE) for the test sets over all ten runs of the cross-validation. Note that NMSE is always a positive quantity and that values smaller than one denotes an improvement wrt the simplest predictor, i.e. the sample average.

We perform three experiments:

1. The first experiment deals with the centralized configuration where all sensor measurements are transmitted with no processing to the data mining server. We perform a recursive PCA with an increasing number m of principal components. The NMSE results for $m = 2, \ldots, 7$ after that N_{tr} samples have been processed are reported in Table 1. This table serves as reference for the results obtained in the modular case.

2. This experiment assesses the prediction accuracy of 6 uniform partitions $\{\mathcal{P}^{(2)}, \ldots, \mathcal{P}^{(7)}\}$ of the whole

m	1	2	3	4	5	6
NMSE	0.782	0.363	0.257	0.223	0.183	0.196

Table 1: Centralized configuration: NMSE for different numbers m of principal components. NMSE is evaluated through a 10-fold cross-validation procedure after N_{TR} examples have been processed.

	$\mathcal{P}^{(2)}$	$\mathcal{P}^{(3)}$	$\mathcal{P}^{(4)}$	$\mathcal{P}^{(5)}$	$\mathcal{P}^{(6)}$	$\mathcal{P}^{(7)}$
NMSE	0.140	0.118	0.118	0.118	0.116	0.114

Table 2: Modular configuration: NMSE obtained with six different partitions of the $S = 2372$ sensors. NMSE is evaluated through a 10-fold cross-validation procedure after N_{TR} examples have been processed.

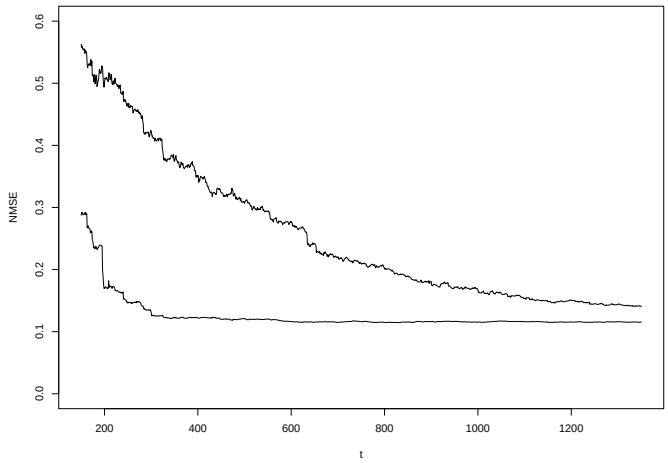

Figure 4: The sequential evolution of the NMSE for the partition $\mathcal{P}^{(2)}$ (upper line) and $\mathcal{P}^{(5)}$ (lower line). The NMSE at each instant t is an average over the ten runs of the cross validation procedure. The x-axis report the number of observations taken into consideration.

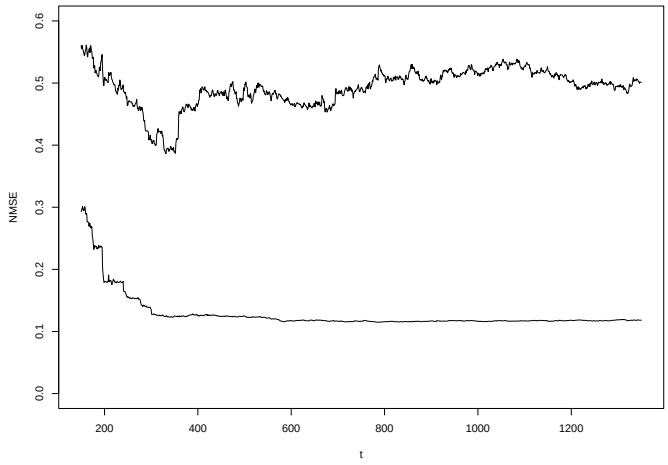

Figure 5: The sequential evolution of the NMSE for the partition $\mathcal{P}^{(2)}$ (upper line) and $\mathcal{P}^{(5)}$ (lower line) in front of random malfunctioning. The NMSE at each instant t is an average over the ten runs of the cross validation procedure. The x-axis report the number of observations taken into consideration.

set of sensors. Table 2 reports the NMSE of the test set after that N_{tr} samples have been processed. Since the data processing is carried out in a recursive fashion, we can also analyze the evolution of the NMSE for the test set, as more observations reach the mining server. Figure 4 reports the evolution of the Normalized Mean Square Error (NMSE) for the partition $\mathcal{P}^{(2)}$ and the partition $\mathcal{P}^{(5)}$.

3. The third experiment assesses the degradation of the accuracy in front of random malfunctioning of the sensors. We consider the 6 partitions assessed in the first experiment and we analyze the evolution of the NMSE when we simulate the occurrence of sensor faults. We assume that at each observation there is a 10% probability that a sensing unit is switched off and a 1% probability that one of the aggregating unit becomes out of order. Note that each malfunctioning is permanent and a sensing unit which breaks down cannot be reactivated. In data analysis terms, this is equivalent to the disappearance of some input variables for the recursive PCA procedure and the lazy learning algorithm, respectively. The NMSE after the processing of N_{tr} observations is reported in Table 3. These figures as well as the evolution of NMSE during the processing of N_{tr} observations (Figure 5), show that the degradation is negligible if the ratio between failure probability and unit number is not too high. This limitation in the system reliability is illustrated in the case of partition $\mathcal{P}^{(2)}$ (Figure 5) for which a dramatic loss in accuracy is observed. By increasing the number of clusters, one increases the robustness of the architecture, which thanks to the recursive feature of its components is able to react accordingly to faults and malfunctioning.

	$\mathcal{P}^{(2)}$	$\mathcal{P}^{(3)}$	$\mathcal{P}^{(4)}$	$\mathcal{P}^{(5)}$	$\mathcal{P}^{(6)}$	$\mathcal{P}^{(7)}$
NMSE	0.501	0.132	0.119	0.116	0.116	0.117

Table 3: NMSE obtained with six different partitions of the $S = 2372$ sensors in front of random malfunctioning. NMSE is evaluated through a 10-fold cross-validation procedure after N_{TR} examples have been processed.

7 Conclusions and future work

The paper proposed an adaptive methodology to mine data in large sensor networks. Previous works focused mainly on addressing issues of energy and transmission bandwidth reduction independently of the sensing task to be performed. This paper advocates the idea that the structure of the processing architecture of a sensor network must take into account also criteria related to the accuracy and quality of the data mining task. This means that the organization of the same sensor network may change according to the type of sensing task (e.g. classification or prediction) and the required quality and precision.

The results shown in this paper are preliminary but open the way to further research whose main lines can be summarized as follows:

- Extensions to non simulated data.
- Combination of accuracy based criteria with energy related cost functions in a multi-criteria configuration procedure.
- Assessment of more sophisticated clustering policies to partition the sensor field.
- Combination of spatial and temporal local correlations.
- Test of nonlinear compression techniques, like ICA.
- Adoption of sequential robust estimation techniques.

References

[1] M. Birattari, G. Bontempi, and H. Bersini. Lazy learning meets the recursive least-squares algorithm. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *NIPS 11*, pages 375–381, Cambridge, 1999. MIT Press.

[2] C. M. Bishop. *Neural Networks for Statistical Pattern Recognition*. Oxford University Press, Oxford, UK, 1994.

[3] G. Bontempi. *Local Learning Techniques for Modeling, Prediction and Control*. PhD thesis, IRIDIA- Université Libre de Bruxelles, 1999.

[4] G. Bontempi, M. Birattari, and H. Bersini. Local learning for iterated time-series prediction. In I. Bratko and S. Dzeroski, editors, *Machine Learning: Proceedings of the Sixteenth International Conference*, pages 32–38, San Francisco, CA, 1999. Morgan Kaufmann Publishers.

[5] G. Bontempi, M. Birattari, and H. Bersini. A model selection approach for local learning. *Artificial Intelligence Communications*, 121(1), 2000.

[6] S. Goel and T. Imielinski. Prediction-based monitoring in sensor networks: Taking lessons from mpeg. *ACM Computer Communication Review*, 31(5), 2001.

[7] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[8] A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, 2001.

[9] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[10] L. Subramanian and R. H.Katz. An architecture for building self-configurable systems. In *IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC 2000)*, 2000.

[11] B. Yang. Projection approximation subspace tracking. *IEEE Transactions on Signal Processing*, 43(1):95–107, 1995.