Unsupervised and supervised compression with principal component analysis in wireless sensor networks

Yann-Aël Le Borgne
yleborgn@ulb.ac.be

Gianluca Bontempi
gbonte@ulb.ac.be

ULB Machine Learning Group
Computer Science Department
Université Libre de Bruxelles (U.L.B.)
1050 Brussels - Belgium
Tel: +32-2-6505594
Fax: +32-2-6505609

ABSTRACT

This paper shows that the Principal Component Analysis, a compression method widely used in statistical anaylsis and image processing, can be efficiently implemented in a network of wireless sensors. The proposed scheme proves to be particularly suitable to sensor networks as it allows to reduce the network load while retaining a maximum amount of variance from sensor measurements. We present two operating modes, unsupervised and supervised, allowing (i) to extract a maximum of variance while keeping the network load bounded, and (ii) to reduce the network load while keeping the approximation error bounded, respectively. We assess the efficiency of the proposed approach in a realistic wireless sensor network deployment for temperature monitoring.

Categories and Subject Descriptors

C.2.4 [**Computer Systems Organization**]: Computer - Communication Networks—*Distributed applications, distributed databases*; G.1.2 [**Mathematics of Computing**]: Numerical analysis—*Least squares approximation*; G.3 [**Mathematics of Computing**]: Probability and statistics—*Correlation and regression analysis*

General Terms

Wireless sensor networks, Principal component analysis, In-network compression

Keywords

WSN, PCA, compression

1. INTRODUCTION

We consider in this paper wireless sensor network (WSN) applications where sensor measurements are collected at regular time instants, and transmitted by a routing tree to a specific network component called the *sink* [1]. Such applications include for example long term environmental monitoring, structural monitoring, battlefield surveillance, etc.. [3, 18, 17].

The routing tree allows measurements of sensor nodes far away from the sink to be relayed by intermediate sensor nodes so that they eventually get delivered to the sink. Networks relying on routing trees are instances of *multi-hop* networks, meaning that some packets are relayed by intermediate network nodes [2, 4]. We assume in this paper that there exists a routing layer suitable for data aggregation, which synchronizes transmissions between nodes in such a way that sensor nodes deeper in the tree sends their measurements before their parents. An illustration of this routing scheme is given in Figure 1. Research projects on query processing architectures over sensor networks, such as those developed at UC Berkeley (TinyDB and TAG projects) [13, 14], Cornell University (COUGAR project) [19] or EPFL (Dozer) [6], have provided the WSN community with such routing layers. Their advantages are twofold. First, they maximize the sleeping time of sensor nodes by synchronizing the transmissions along the routing tree. Second, they allow to aggregate data along the tree, so as to provide the sink with a summary of the measurements collected in the sensor field. Schemes allowing to efficiently extract summaries such as the mean, the median, the quantiles, or the contours have been proposed using this kind of synchronized routing layer in [20, 13, 16, 8].

In this paper, we show that the *principal component analysis* (PCA) [15], a classic technique in statistical data analysis for data approximation and compression, can be efficiently implemented in a WSN relying on a synchronized routing layer. The PCA allows to determine a coordinate system called the *principal component basis*, in which sensor measurements are uncorrelated. As in most cases there exists high spatial correlations between sensor measurements, good approximations to sensor measurements can be obtained by relying on few principal components.

[1]In the WSN literature, the *sink* is the network component that gathers all sensor measurements, and is usually considered to benefit from higher computational resources than the sensor nodes (e.g. a desktop computer)

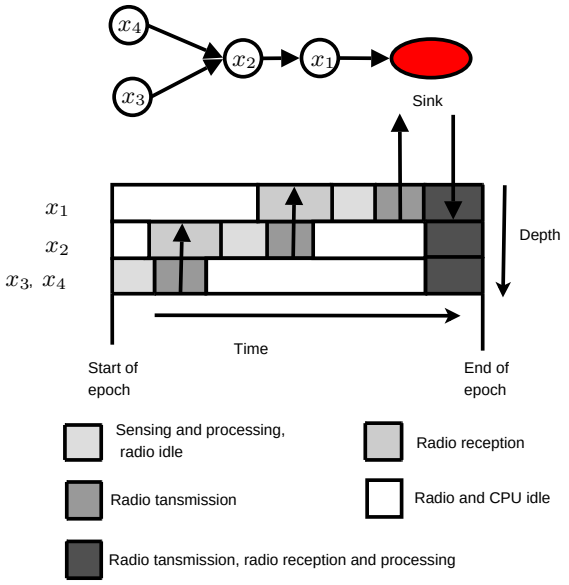

Figure 1: Activities carries out by sensors depending of their depth in the routing tree *(adapted from [13])*. Transmissions are synchronized for optimizing energy savings. The last stage involves all sensors and allows unsynchronized synchronization (for sensor discovery e.g.).

The procedure proposed is a two-stage process, in which a set of N measurements is first collected from the whole set of sensors. In a second stage, a set of q *principal components* are computed at the sink, and distributed in the network. Each sensor node needs only to be aware of a vector of weights, the size of the number of retained principal components, for the coordinates in the principal component basis to be computed.

There exists tradeoffs between the accuracy of the measurement approximations, the network load, and the network lifetime. We provide an analysis of these tradeoffs for two operating modes, involving unsupervised and supervised compression. In the unsupervised mode, the approximations obtained at the sink are expected to retain the maximum amount of variance from the sensor field while keeping the network load to a user defined threshold (e.g. the maximum number of packets per sensor per epoch must not exceed three). In the supervised mode, an additional feedback mechanism is set up, that allows to guarantee that approximations to sensor measurements obtained at the sink lie within a user defined threshold (e.g. approximated temperature must lie within $\pm 1^{\circ}C$ of the real measurements). The number of principal components is estimated in such a way that the network load is minimized.

The article is structured as follows. We introduce in section 2 the notation and formulation of PCA, together with the in-network aggregation scheme of sensor measurements on principal components. Section 3 details the unsupervised and supervised compression modes, together with an analysis of the tradeoffs involved between the number of components used, the accuracy of the measurement approximations, and the network load. We provide in section 4 a set of experimental results based on a real world data set of temperature data that illustrate the benefits of the proposed approach. Section 5 contains a discussion on the proposed schemes. The conclusions are summarized in section 6.

2. PRINCIPAL COMPONENT ANALYSIS

2.1 Notations

Let $\mathcal{X} = \{x_1, x_2, ...x_p\}$ be a set of p sensors and let $\mathcal{T} = \{1, 2, 3, ...\}$ be a discretized time domain accounting for the sampling period at which the sensor measurements are collected. The sampling period is also referred to as *epoch*.

Each sensor generates a stream of data. Let $x_i[t]$, $1 \leq i \leq p$, be the measurement taken by sensor i at time $t \in \mathcal{T}$ and let $\mathbf{x}[t] = (x_1[t], x_2[t], ..., x_p[t]) \in \mathbb{R}^p$ be the column vector of measurements taken in the sensor field at time t. Let $X_{p \times N}$ be a matrix with elements $x_{it} = x_i[t]$, containing columnwise N observations of the sensor field $\mathbf{x}[t]$, $1 \leq t \leq N$. Finally, let $\bar{\mathbf{x}}[\mathbf{t}] = \frac{1}{N} \sum_{t=1}^{N} \mathbf{x}[t]$ be the $X_{N \times p}$ mean vector columnwise.

2.2 Formulation

Principal Component Analysis (PCA) is a classic technique in statistical data analysis, data compression, and image processing [15, 10]. Given $q \leq p$ and a set of N centered multivariate measurements $\mathbf{x}[t] \in \mathbb{R}^p$, it aims at finding a basis of q orthonormal vectors $\{\mathbf{w_k}\}_{1 \leq k \leq q}$ of $\mathbb{R}^p$, such that the mean squared distances between $\mathbf{x}[t]$ and their projections $\hat{\mathbf{x}}[\mathbf{t}] = \sum_{k=1}^{q} \mathbf{w_k} \mathbf{w_k}^T \mathbf{x}[t]$ on the subspace spanned by the basis $\{\mathbf{w_k}\}_{1 \leq k \leq q}$ is minimized[2]. The corresponding optimization function

$$\begin{aligned} J_q(\mathbf{x}[t], \mathbf{w_k}) &= \frac{1}{N} \sum_{t=1}^{N} ||\mathbf{x}[t] - \hat{\mathbf{x}}[\mathbf{t}]||^2 \\ &= \frac{1}{N} \sum_{t=1}^{N} ||\mathbf{x}[t] - \sum_{k=1}^{q} \mathbf{w_k} \mathbf{w_k^T} \mathbf{x}[t]||^2 \end{aligned} \quad (1)$$

under the constraint of orthonormal $\{\mathbf{w_k}\}_{1 \leq k \leq q}$ can be solved using the Lagrange multiplier technique [9]. The minimizer of Formula (1) is the set of the q first eigenvectors $\{\mathbf{w_k}\}$ of the correlation matix XX^T, ordered for convenience by decreasing eigenvalues λ_k. Eigenvalues quantify the amount of variance conserved by the eigenvectors, and their sum equals the total variance of the original set of observations X, i.e.:

$$\sum_{k=1}^{p} \lambda_k = \frac{1}{N} \sum_{t=1}^{N} ||\mathbf{x}[t]||^2 \quad (2)$$

The proportion P of retained variance with the first q principal components, which characterizes the accuracy of the approximation, is expressed by:

$$P(q) = \frac{\sum_{k=1}^{q} \lambda_k}{\sum_{k=1}^{p} \lambda_k} \quad (3)$$

It is common practice in signal processing and data analysis to retain the first q eigenvectors such that $P(q) = 0.95$, i.e. to conserve 95% of the variance of the original signal.

[2]measurements are centered so that the origin of the coordinate system coincides with the centroid of the set of measurements. This translation is desirable to avoid a biased estimation of the basis $\{\mathbf{w_k}\}_{1 \leq k \leq q}$ of $\mathbb{R}^p$ towards the centroid of the set of measurements.

Ranging columnwise the set of vectors $\{\mathbf{w_k}\}_{1\leq k\leq q}$ in a $W_{p\times q}$ matrix, approximations $\hat{\mathbf{x}}[\mathbf{t}]$ to $\mathbf{x}[t]$ in $\mathbb{R}^p$ are obtained by

$$\hat{\mathbf{x}}[\mathbf{t}] = WW^T\mathbf{x}[t] = W\mathbf{z}[\mathbf{t}] \quad (4)$$

where

$$\mathbf{z}[\mathbf{t}] = W^T\mathbf{x}[t] \quad (5)$$

denote the column vector of coordinates of $\hat{\mathbf{x}}[\mathbf{t}]$ in $\{\mathbf{w_k}\}_{1\leq k\leq q}$, also referred to as *principal coordinates*.

Example: In figure 2 are plotted (circles) a set of $N = 50$ observations involving three data sources $x_1[t]$, $x_2[t]$ and $x_3[t]$. The correlation between $x_1[t]$ and $x_2[t]$ is high, whereas $x_3[t]$ measurements were drawn independently. The set of vectors $\{w_1, w_2, w_3\}$ of the principal component (PC) basis were computed, together with the two-dimensional subspace spanned by $\{w_1, w_2\}$. The projections of the original measurements on the subspace are represented by crosses, and illustrate that this set of measurements can be well approximated by the two first PCs, as there exists strong correlations between x_1 and x_2.

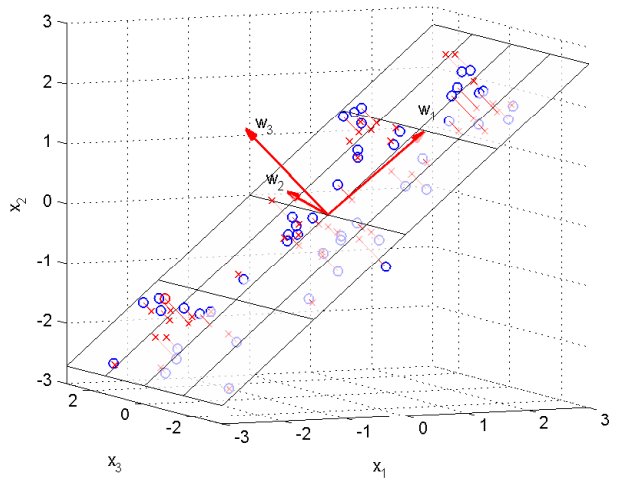

Figure 2: Illustration of the transformation obtained by the principal component analysis. Circles give the original observations, and crosses their approximations on the two-dimensional subspace spanned by the two first principal components.

2.3 Compression strategies

2.3.1 Initialization stage

In a first stage, we assume that a set of N observations are gathered at the sink from the whole sensor field. We assume that a synchronized routing layer has been set up, and that measurements from sensors out of communication range of the sink have their measurements relayed by the means of the routing layer. This data collection mode is referred to as the *default data collection mode*, to denote the absence of compression.

This stage allows to build, after N epochs, an $X_{p\times N}$ matrix from which q principal components are extracted. The number N of observations to gather should be chosen such that the average correlations between sensor measurements are well captured, in order to properly identify the principal components. This will be discussed more in depth in section 4 and 5.3. The choice for q will be addressed in section 3.2 and 3.3.

Once computed, the $W_{p\times q}$ matrix of principal components is flooded in the network by the means of the routing tree, from the root down to the leaves. Each sensor x_i, $1 \leq i \leq p$, only retains the i-th row of the matrix. The network can from this moment switch to any of the two compression strategies described hereafter.

2.3.2 Unsupervised compression

The central point of the compression strategies proposed in this paper is that, at each epoch, the projection $\mathbf{z}[\mathbf{t}]$ (Formula 5) can be computed as data traverses the routing tree. Letting w_{ik} be the i-th element of the k-th principal component, the k-th principal coordinate $z_k[t]$ at time t is obtained by the following scalar product:

$$z_k[t] = \sum_{i=1}^{p} x_i[t] * w_{ik} \quad (6)$$

Assuming that sensor x_i has available the set of q elements $\{w_{ik}\}$, $1 \leq k \leq q$, the vector of coordinates z_t can be easily computed along the routing tree. The aggregation process is illustrated in Figure 3 (left) for a network of four nodes, in which the coordinate of the first principal coordinate is aggregated along a routing tree of depth three. The notation $z_k^{\{S\}} = \sum_{i\in S} x_i[t] * w_{ik}$ is used for detailing the progression of the scalar product along the routing tree. The set $\{S\}$ is the set of sensors whose measurements have already been aggregated. The elements available at sensors x_i are reported as vectors on the side of the sensor symbol. The root of the routing tree is the last step of the aggregation process, where we have

$$z_k^{\{1,2,\ldots,p\}} = \sum_{i=1}^{p} x_i[t] * w_{ik} = z_k[t] \quad (7)$$

The transformation of the vector of coordinates z_t back to the original basis can then be achieved at the base station using Formula (4) to get an approximation $\hat{x}_t$ of the measurements over the whole sensor field.

The main benefit of this approach is that the set of q coordinates $z_k[t]$ in the principal component basis, $1 \leq k \leq q$, can be delivered to the base station **with a constant packet size for each traversed node**. As in most cases a good approximation can be obtained by relying on few principal components, the proposed scheme can allow to significantly reduce the network load at the root of the routing tree. This scheme is dubbed *unsupervised* as approximations obtained at the sink are not checked against actual measurements collected by the sensors, and will be analyzed in more details in section 3.2.

2.3.3 Supervised compression

The computation required to get the approximation $\hat{x}_i[t]$ for the ith sensor, $1 \leq i \leq p$, requires the knowledge of the q

principal coordinates z_k and the q elements $\{w_{ik}\}$, $1 \leq k \leq q$ (cf Formula 4):

$$\hat{x}_i[t] = \sum_{k=1}^{q} z_k[t] * w_{ik} \quad (8)$$

The elements $\{w_{ik}\}$ are already assumed to be available at each sensor for computing the principal coordinates (cf Formula 6 above). Therefore, it is only sufficient to send back to the routing tree the k principal coordinates to have each sensor compute the approximation that was made at the sink.

This additional stage allows to verify that the approximations are within some user defined ϵ of the true measurements. Whenever the condition is not met for a sensor, it sends the true measurement to the sink. **This scheme guarantees that all data eventually obtained at the sink are within $\pm\epsilon$ of their the true measurements**.

This second scheme is dubbed *supervised* compression, and will be analyzed in more details in section 3.3.

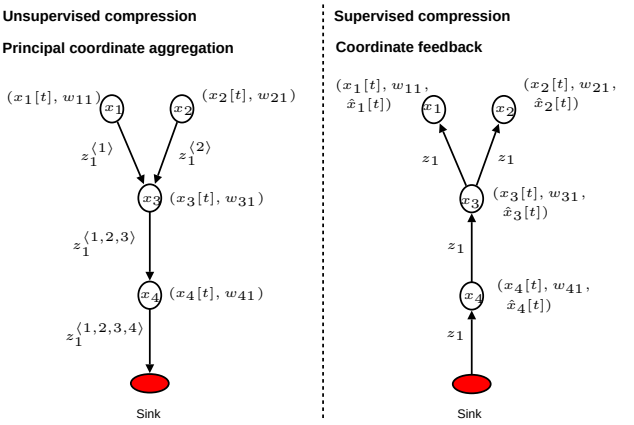

Figure 3: Illustration of the aggregation process where the coordinate of the first principal component is computed along a routing tree of depth three in network of four nodes (left) and then fed back in the network to allow sensors to recover approximations (right).

2.3.4 A note on the mean

We assumed from section 2.2 that the sensor measurements had zero mean, so that their centroid coincided with the origin of the coordinate system. This assumption can be relaxed easily, if the mean is subtracted by the sensor prior to the aggregation of its value (in Formula 6), add added back after the computation of the approximation (in Formula 8). The mean value of the measurements collected by a sensor can be either computed at the sink after the N epochs, or computed in a recursive manner by the sensor node.

3. TRADEOFF ANALYSIS

The previous section presented how some of the computation required by the PCA could be distributed in the network, and provided two compression strategies. This section discusses how these compression strategies can be relied on in practice, by analyzing the tradeoffs involved between the network lifetime, the network load, and the compression accuracy.

3.1 Metrics

In the following, we assume that each node has initially the same energy budget, and that one packet is required to transmit a piece of information over the network (either a single sensor measurement or a coordinate). We also assume, for the sake of simplicity, that there is no sensor failure, and that all packets are delivered to the sink. Relaxation of these assumptions will be discussed in section 5.

We chose as metrics for the network lifetime and the network load **the time-to-first failure (TTFF)** and **the maximum network load**, respectively. The TTFF is a commonly used metric for characterizing network lifetime, and defines the duration of time before any node in the network runs out of its battery energy. In wireless sensor networks, it can be to a first approximation linked to the sensor that has the maximum network load[3], as the radio communication is one of the most energy consuming task for a wireless sensor node [**?**].

In the case of a routing tree, the node with the maximum network load is the root, as it is the last node to be traversed before the packets reach the base station [4]. Note that for a tree of size p, the root node is required to transmit p packets per epoch, and that any additional node increases the maximum network load, thereby decreasing the TTFF.

The accuracy metrics are **the percentage of retained variance** for the unsupervised compression mode, and the **absolute error** for the supervised compression mode. They will be detailed in sections 3.2 and 3.3, respectively.

3.2 Unsupervised compression

In the unsupervised compression mode, the tradeoff between accuracy, network load and network lifetime can be addressed by Formula 3, which relates the amount of retained variance by the PCA to the number q of principal components used. The network load is q, as q packets traverse the root of the routing tree at each epoch.

The function $P(q)$ increases monotonically with q, as increasing the number of principal components necessarily increases the amount of retained variance. When data sources are uncorrelated and have the same variance, this function increases linearly with q. Interestingly, when data sources are correlated, the percentage of conserved variance typically first increases sharply, to reach an inflexion point after which the gain in retained variance is much lower as q is increased. The sharp increase corresponds to the components that support the signal of interest, whereas components that provide little gain account for the measurement noise and can be dismissed.

[3]Such a node is often referred to as a *hot spot* in the networking literature

[4]We assume for the sake of simplicity that there is no disjoint trees in the routing structure of the network.

Figure 6 in section 4 provides an illustration of the profile obtained for $P(q)$ for the temperature dataset considered in the experimental section.

This tradeoff can be addressed in two different manners, either by fixing the number q of principal components, or by fixing a percentage of variance to retain. The former case is particularly suitable if the network load is bounded. In this case, the number of components can be fixed to the highest value that complies with the network load limit. The proposed scheme is hence the optimal scheme regarding the amount of retained variance. The latter case is more suitable if the amount of variance to keep for the requirements of the application is known. In this case, the proposed scheme will be optimal with respect to the network load incurred (and consequently optimal in maximizing the network lifetime). In practice, the amount of variance required for an application is however rarely known, and is usually arbitrarily set between 90% and 99%.

3.3 Supervised compression

In the supervised compression mode, the tradeoff between accuracy, network load and network lifetime cannot be analytically addressed as in the previous section.

First, at time t, all sensors whose aprroximations are not within the ϵ error tolerance fixed by the user (cf. Formula 8) are required to send their actual measurements to the sink. Their number therefore depends on the time, on the number q of principal components, and also on the user defined error tolerance ϵ. Letting $U(q, \epsilon, t)$ denote for this quantity, we have

$$U(q, \epsilon, t) = \sum_{i=1}^{N} 1_{(|x_i - \hat{x}_i| > \epsilon)}(x_i) \quad (9)$$

where 1() is the binary indicator function. The network load at each time instant is therefore given by

$$L(q, \epsilon, t) = 2q + U(q, \epsilon, t) \quad (10)$$

Additionally, a minimum number of $2q$ packets will be transmitted through the root of the tree at each epoch: q packets as the q principal coordinates are computed along the tree, and q packets as the principal coordinates are routed back from the sink to the tree.

The tradeoff involved in the supervised compression mode is as follows. For ϵ fixed, the number of updates decreases as q increases, given that the approximations are necessarily closer to the true measurements as was pointed out in the previous section. As the number of updates is typically high for low q, the cost given by Formula 10 may therefore show a decreasing stage before monotonically increasing as q increses.

For q fixed, the number of updates clearly monotonically decreases with the error tolerance ϵ as the higher the error tolerance, the lower the number of updates, and conversely.

Example of profiles for this tradeoff are reported in Figure 9 (right) in section 4.

4. EXPERIMENTAL RESULTS

4.1 Data

Experiments were carried out using a set of temperature readings obtained from a 54 Mica2Dot sensor deployment at the Intel research laboratory at Berkeley [1]. Data was originally sampled every thirty-one seconds, and the dataset available from the data webpage associates each measurement collected at the base station to a sensor node ID and a timestamp. We selected temperature data from a set of eight consecutive days, where measurements had few missing values. Using principal components aggregation, missing values need to be handled with special care, as is discussed in section 5. In the present case, missing values were linearly interpolated from other data during a preprocessing stage where data was discretized in thirty second intervals. We mention that this interpolation does not affect the results presented hereafter. The sensors 5 and 15 were removed as they did not provide any measurement. After preprocessing, the dataset contained a trace of 23040 readings from 52 different sensors.

Examples of temperature profiles and dependencies between measurements are reported in Figures 4 and 5, respectively. The sensors 21 and 49 were the least correlated ones over that time period, with a correlation coefficient of 0.59. They were situated on opposite sides of the laboratory. Sensors 49 and 47 were side by side, and their measurements exhibited a particularly high level of dependency (the correlation coefficient is 0.99). Temperature over the whole set of data ranged from about 15°C to 35°C.

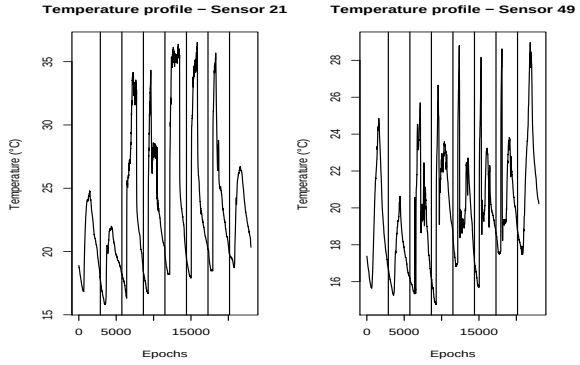

Figure 4: Temperature measurements collected by sensors 21 and 49 over an eight day period. Sensors 21 and 49 were the least correlated ones during that period.

4.2 Training and testing

Let us recall that unsupervised and supervised compression modes are two-stage processes implying (i) a default data collection stage over N time instants that fills a matrix of observation $X_{p \times N}$ whose principal components are computed, and (ii) an approximate data collection stage where new measurements, that were not used for the computation of the principal components (PCs), are approximated.

To account for this two-stage process, the set of observations was partitioned in two parts. The first four days of observations (i.e. 11520 epochs) were used to compute the measurement mean and principal components. The last four days

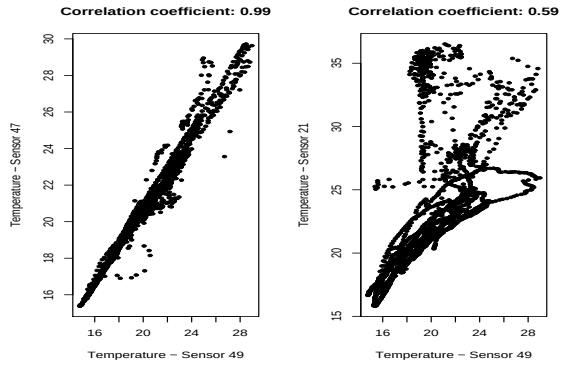

Figure 5: Profiles of the dependencies between sensor measurements for (nearly) the most correlated (left) and least correlated (right) pairs of sensors.

of observations were used to test the compression schemes. These two subsets are referred to as *training* and *test* sets in the following.

As reported in Table 4.2, it is actually not necessary to use the full set of observations gathered during the first four days to get satisfying estimates for the principal components. Table 4.2 reports the percentage of variance retained on the test set for different number of principal components (PCs), and different sizes of training sets.

The accuracy of the compression neatly increases as the size of the training sets grows from 6 hours of observations to 12 hours. Augmenting further the training set to one day up to four days allows to get slightly more accurate approximations.

These results stem from the fact the temperature measurements exhibit daily cyclic patterns. A set of temperature measurements spanning a one day period therefore provides a representative set for extracting the correlations existing among the sensors, and all the results reported in section 4.3 and 4.4 were obtained using one day of data (2880 epochs).

Table 1: Percentage of variance retained on the test set for different number of principal components (PCs) and different sizes of training sets (%).

	4 days	3 days	2 days	1 day	12 hrs	6 hrs
1 PC	82	81	81	81	81	69
4 PCs	95	93	92	90	91	76
8 PCs	98	97	97	96	95	87

4.3 Unsupervised compression

4.3.1 *Tradeoff network load - Accuracy*

An important practical issue in running the unsupervised compression mode is to properly estimate the amount of retained variance by the principal components. We pointed out that the retained variance could be estimated by the means of the training set using Formula 3. However, such estimate is in practice optimistic as it is based on the data from which the principal components were computed. The performances obtained on new data, independent of the training set, are usually lower.

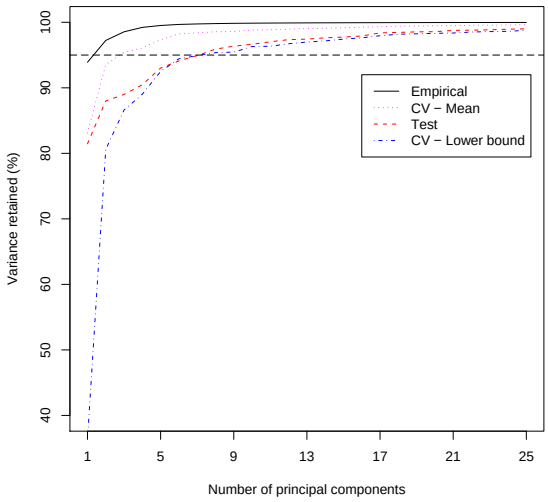

Figure 6: estimated and actual percentages of retained variance on the test set for an increasing number of principal components, using different estimation methods.

The assessment of the accuracy of a learning process on new data is a typical machine learning issue, which can be addressed by relying on K-cross-validation (K-CV). The rationale of K-CV is to divide a training set in K subsets, to use K-1 subsets for the learning, and to estimate the accuracy on the remaining subset of data.

After simulating K times the learning process, K-CV provides an estimate of the mean of the accuracy criterion considered, together with its standard deviation. A probabilistic bound on the accuracy expected on new data can be derived by computing a confidence interval around the expected mean accuracy.

We suggest to rely on this technique to properly estimate the percentage of retained variance. We report in Figure 6 the estimated and actual percentages of retained variance on the test set for an increasing number of principal components, using different estimation methods. The actual retained variance on the test set is given by the dashed curve (third from the top, referred to as *test*).

The upper continuous curve (referred to as *empirical*) reports the percentage of retained variance $100 * P(q)$ using Formula 3, and is particularly optimistic. The second curve from the top (*CV-mean*) is the estimated mean accuracy obtained by 10-CV. Finally the bottom curve (*CV-lower bound*) is the lower bound of a 95% confidence interval around the 10-CV estimated mean, obtained by relying on the 10-CV estimated variance and a student distribution table.

The CV-lower bound curve provides a good estimate of the

actual accuracy obtained on the test, despite underestimating the accuracy for low numbers of principal components.

From Figure 6, we observe that the percentage of retained variance increases very quickly as the number of principal components used increases. The first principal component retains about 80% of the variance, and 4 principal components increase this amount to 90%. The conservation of 95% of the variance require relying on the first eight principal components.

These results shows that appealing accuracies can be obtained with very few components, and illustrate the efficiency of relying on the principal components when collecting spatially correlated data.

4.3.2 *Approximations to original data*

Fig. 7 and 8 illustrate the approximations obtained on the test set for the sensors 21 and 49, using one, four and eight principal components.

These sensors, as mentioned in section 4.1, are the least correlated, and follow different patterns during the day. More particularly, the temperature obtained for sensor 49 seems to be artificially stabilized around 20 Celsius degrees during the first three days. We conjecture that the air conditioning was switched on during these periods.

We notice that a single principal components provide a rough approximation, which cannot account for the specifities of some sensor variations. For example, the stabilization of the temperature in the area close to sensor 49 is not rendered after the compression.

We also note that increasing the number of principal components may not improve approximations in the same manner. For example, while passing from one to four PCs does not improve the approximations obtained for sensor 21, it provides significant improvements for sensor 49. Contrarily, passing from four to eight PCs clearly improves approximations for sensor 21, whereas it is of less benefit for approximations obtained for sensor 49.

4.4 Supervised compression

4.4.1 *Absolute error over time*

We reported in Figure 9 (left) the absolute error obtained on the test set for the sensor 49 for different number of PCs. Note that this figure are dual to Figure 8, but provides a better illustrative support to the supervised compression mode.

Given a sensor and a user-defined error tolerance ϵ, all epochs at which the absolute error is larger than ϵ will require an additional transmission to correct the approximation made at the sink. Here, an $\epsilon = 1°C$ was set (horizontal line).

The overall number of errors is reported in Figure 9 (center) for different numbers of principal components, as boxplots. It can be observed that these numbers decrease very quickly as the the number of PCs increases. In the case of the temperature data studied in this section, we observe that these updates are however not well spread out over time (from Figure 9 (left) and may cause punctually high network loads at

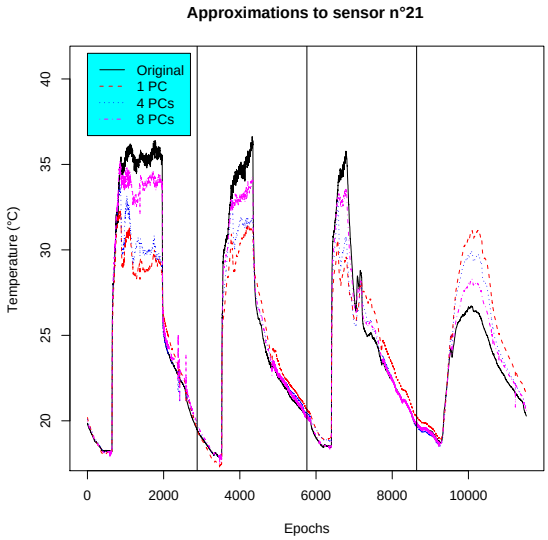

Figure 7: Approximations obtained on the test set for the sensor 21, using one, four and eight principal components.

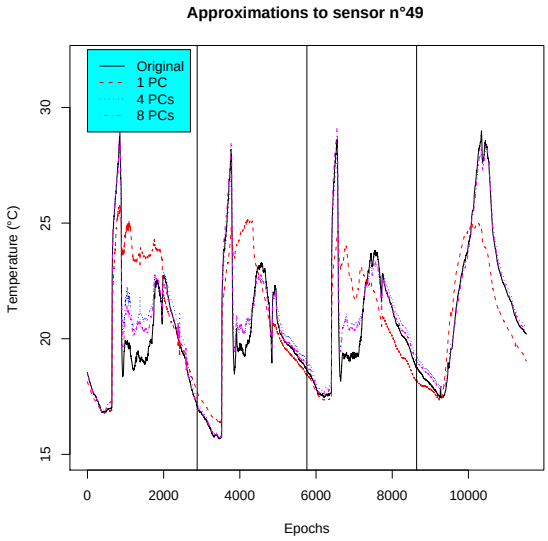

Figure 8: Approximations obtained on the test set for the sensor 49, using one, four and eight principal components.

the routing tree (but still less or equal to the default data collection mode).

4.4.2 Tradeoff network load - accuracy

We varied the error tolerance ϵ from $0°C$ to $10°C$, and reported in Figure 9 (right) the number of packets transmitted at the root of the routing tree as the number of solicited PCs increases.

Two extreme cases can be observed. First, when the error tolerance is set to zero, meaning no tolerance for approximations (upper line), the optimal number of PCs is 0, and the associated network load 52. This comes down to the default data collection mode.

The second extreme case happens if the error tolerance is wider than the measurement range (lower line), in which case the network of sensors need not be used. The optimal number of PCs is therefore also zero.

In between these two extremes, the optimal number of principal components for the temperature data considered is around two and eight.

As for the supervised compression mode, an important practical issue is to determine, from a training set, the optimal number q of components to use. We relied for this purpose on the K-cross validation scheme described in the previous section. Table 4.4.2 reports in its second column (CV best) the number of components retained with the 10-CV for varying ϵ, together with the estimated average network load at the root of the routing tree. The first column (Test best) reports the true optimal number of PCs for the test set, with the network loads incurred. The third column (Test obtained) reports the network load incurred on the test set using the number of PCs estimated by the 10-CV. These results show that 10-CV proved to be suitable in finding near optimal solutions.

	Test best	CV best	Test obtained
$\epsilon = 0°C$	0 - 52.00	0 - 52.00	0 - 52.00
$\epsilon = 0.1°C$	1 - 48.57	1 - 49.35	1 - 48.57
$\epsilon = 0.25°C$	2 - 39.44	4 - 39.47	4 - 40.64
$\epsilon = 0.5°C$	2 - 27.83	4 - 23.81	4 - 28.12
$\epsilon = 1°C$	2 - 15.63	4 - 13.21	4 - 17.89
$\epsilon = 3°C$	1 - 5.10	1 - 3.20	1 - 5.10
$\epsilon = 10°C$	0 - 0.38	0 - 0.02	0 - 0.38

5. DISCUSSION AND RESEARCH TRACKS

The two PCA based compression modes proposed in this paper were shown to be well suited to the wireless sensor network framework. We discuss in this section issues that have been so far left aside, and open the approach to possible extensions.

5.1 Packet losses

Packet losses entail an incomplete computation of the projections on principal components. This may corrupt the reconstruction of the whole sensor field measurements at the base station. Therefore, while a packet loss in the default data collection mode merely causes a missing measurement, it can jeopardize the whole set of measurements in principal components aggregation.

Packet loss is therefore an important issue that must be handled with care. The most straightforward solution is to notify the base station, upon the delivery of the coordinates, of sensors ID whose contributions were not received. This allows the base station to properly reconstruct the sensor field measurements, except for those that where missing (as in the default data collection mode). Note that this incurs additional network load. At the same time, it provides the base station with the IDs of malfunctioning nodes or network links, which could be desirable for maintenance purposes.

5.2 Erroneous measurements

Erroneous measurements are caused by sensor malfunctioning, and are measurements that do not reflect the physical quantity monitored. As for packet losses, they corrupt the computation of the projections on the principal components. Unlike packet losses however, they may be difficult to detect, depending on the dynamic of the phenomenon monitored.

Simple rules may be set up to prevent their integration into the computation of the princpal coordinates (such as temperature outside the range $[-20; 60]$ should be reported as missing data). More sophisticated rules for predicting measurements on the basis of expected temporal or spatial correlations could be considered.

In the case of supervised compression, incorrect computation of the principal coordinates would eventually be detected as approximations are checked against the real measurements. However, it may be that a single erroneous or missing value cause the whole set of sensors to send an update. In a general manner, we believe that the addition of erroneous or missing measurement strategies could provide interesting improvements to the proposed compression schemes.

5.3 Training set size

An important parameter of the proposed approach is the number of observations N that are required to be collected from the sensor network before proceeding to the principal component analysis. This number should be as low as possible, in order to switch from the default monitoring mode to the approximate one. At the same time, too few observations will provide poor estimates of the principal components, leading to larger errors of approximation. This was illustrated in section 4.1 (cf Table 4.2), with the difference observed in the modelling accuracy between 6 hours and 12 hours of measurements.

It should be stressed however that the important point in collecting observations for extracting the principal components is not so much in the number of observations collected, but in the concordance of the distribution of the measurements collected with that of the future measurements. An important issue is therefore raised by non stationary signals, where the amount of correlation between sensor signals may change over time. Changes in the signal correlation matrix directly affect the directions of the principal components, leading to potentially high and unexpected error rate in the compression. The proposed approaches are not well suited

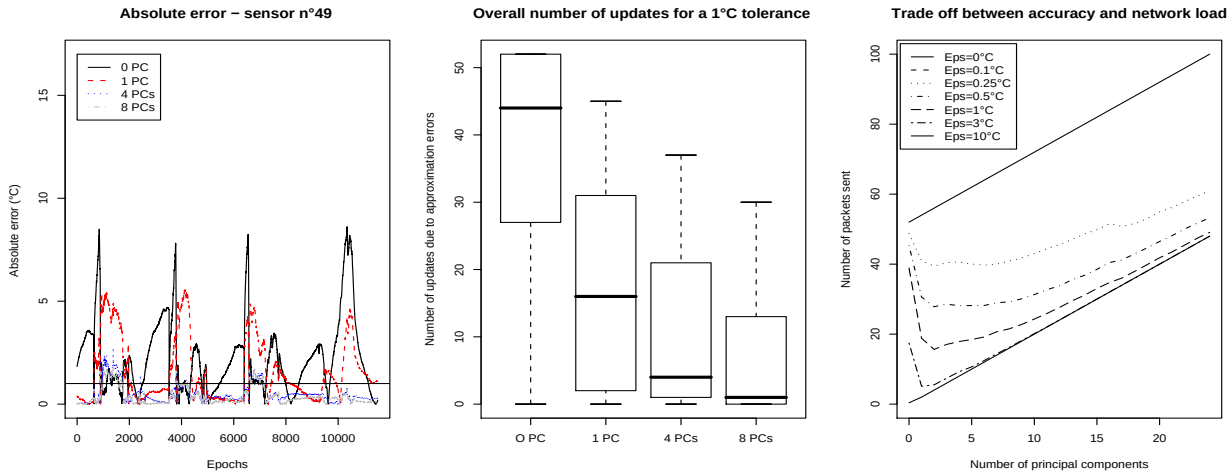

Figure 9: Left: Absolute error on the test set for sensor 49, and for different number of PCs. Center: Boxplots of the numbers of approximations that did not meet the ϵ threshold. Right: Number of packets transmitted at the root of the routing tree, for different error tolerance ϵ, as the number of solicited PCs increases.

to non stationary signals. A possible research direction to handle non stationarity could be to rely on adaptive PCA techniques [9, 7].

5.4 Extension of the aggregation principle

The first research direction to investigate is to extend the proposed framework to the compression of (i) the variations of a signal over time and (ii) different correlated physical quantities captured by a wireless sensor module. The former extension would consist in applying the PCA over space and time, to capture the temporal linear redundancies existing among sensor measurements. Such compression would imply a tradeoff between the network load reduction (due to the compression of measurement over time) and the latency in measurement delivery (as the principal coordinates would be sent every T epochs, $T > 1$). The latter could fusion different types of sensor measurements, typically correlated, such as temperature and humidity. Extending the PCA framework to different physical quantities requires to study how the PCA can be weighted to account for the different amount of variances generated by different physical quantities (e.g. how to combine Celsius degrees with percentages of humidity).

The aggregation principle underlying the compression schemes proposed in this paper are also readily extensible to any basis transformation. Among the basis transformations of interest, we stress that the independant component analysis (ICA), also known as blind source separation, [9] is particularly appealing. ICA aims at determining a basis which not only decorrelates signals, but that also gets them independent. ICA has for example proven particularly efficient in speech processing in separating the set of independent sources composing an audio signal.

Another research track lie in the application of random bases for compressing signal. Of particular interest is the work proposed by [11] where pseudo random bases generated in a distributed manner are shown to probabilistically retain appealing amount of information.

Finally, we mention that measurement transformations provided by PCA or other transformation schemes could also be used as inputs to classification or prediction problems at the network scale. Given the task of recognizing the type of vehicle or the number of heat sources from a network of sensors collecting vibration or temperature measurements, such transformations could be driven to provide dense and informative summaries of the phenomenon monitored. Preliminary work in this direction was discussed in [12, 5].

6. CONCLUSION

In this paper, we proposed two compression schemes based on the removal of spatial linear dependencies between sensor measurements. The paper covered the underlying theoretical elements, showed their straightforward integration in a synchronized routing layer, and illustrate experimentally the benefits of the approach. The proposed schemes were shown to be well suited to multi-hop sensor networks collecting spatially correlated measurements. Some issues were pointed out, and potential solutions addressed. Extensions of the approaches discussed are promising, particularly from the distributed data mining and collaborative signal processing perspectives.

7. ACKNOWLEDGMENTS

This work was supported by the COMP2SYS project, sponsored by the Human Resources and Mobility program of the European Community (MEST-CT-2004-505079).

8. REFERENCES

[1] http://db.csail.mit.edu/labdata/labdata.html. Intel Lab Data webpage.

[2] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 3(3):325–349, 2005.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[4] J. Al-Karaki and A. Kamal. Routing techniques in wireless sensor networks: a survey. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, 11(6):6–28, 2004.

[5] G. Bontempi and Y. L. Borgne. An adaptive modular approach to the mining of sensor network data. In *Proceedings of the Workshop on Data Mining in Sensor Networks, SIAM SDM*, Newport Beach, CA, April 2005.

[6] N. Burri and R. Wattenhofer. Dozer: ultra-low power data gathering in sensor networks. *Proceedings of the 6th international conference on Information processing in sensor networks*, pages 450–459, 2007.

[7] K. Diamantaras and S. Kung. *Principal component neural networks: theory and applications*. John Wiley & Sons, Inc. New York, NY, USA, 1996.

[8] J. Hellerstein, W. Hong, S. Madden, and K. Stanek. Beyond average: Towards sophisticated sensing with queries. *Proceedings of the First Workshop on Information Processing in Sensor Networks (IPSN)*, 2003.

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

[10] I. Jolliffe. *Principal Component Analysis*. Springer, 2002.

[11] S. Kumar, F. Zhao, and D. Shepherd. Collaborative signal and information processing in microsensor networks. *Signal Processing Magazine, IEEE*, 19(2):13–14, 2002.

[12] Y. Le Borgne, M. Moussaid, and G. Bontempi. Simulation Architecture for Data Processing Algorithms in Wireless Sensor Networks. *Proceedings of the 20th International Conference on Advanced Information Networking and Applications-Volume 2 (AINA'06)-Volume 02*, pages 383–387, 2006.

[13] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks. *Proceedings of the ACM Symposium on Operating System Design and Implementation (OSDI)*, 2002.

[14] S. Madden, M. Franklin, J. Hellerstein, and W. Hong. TinyDB: an acquisitional query processing system for sensor networks. *ACM Transactions on Database Systems (TODS)*, 30(1):122–173, 2005.

[15] K. Mardia, J. Kent, J. Bibby, et al. *Multivariate analysis*. Academic Press New York, 1979.

[16] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri. Medians and beyond: new aggregation techniques for sensor networks. *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 239–249, 2004.

[17] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin. Habitat monitoring with sensor networks. *Communications of the ACM*, 47(6):34–40, 2004.

[18] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, et al. A macroscope in the redwoods. *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 51–63, 2005.

[19] Y. Yao and J. Gehrke. The cougar approach to in-network query processing in sensor networks. *ACM SIGMOD Record*, 31(3):9–18, 2002.

[20] J. Zhao, R. Govindan, and D. Estrin. Computing aggregates for monitoring wireless sensor networks. *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pages 139–148, 2003.