Symmetric Monotonic Regression: Techniques and Applications in Sensor Networks

Jennifer L. Wong, Seapahn Megerian[‡], Miodrag Potkonjak[†]
SUNY Stony Brook, [‡]Univ. of Wisconsin, Madison, [†]Univ. of California, Los Angeles

Abstract— **Inter-sensor modeling of data streams is an important problem and an enabler for numerous sensor network tasks such as faulty data detection, missing data recovery, and compression. We have developed a new symmetric monotonic regression (SMR) technique for predicting data at one sensor using data from another sensor or a set of sensors that simultaneously guarantees isotonicity and minimizes an arbitrary form of error for predicting stream X from stream Y and vice versa. Using a simple and fast algorithm, we also developed a lower bound regression (LBR) approach for evaluating the achievable accuracy of regression between the readings at two sensors. SMR often performs very close to the lower bound on a set of collected real-life sensor data. We show how LBR barrier can be outperformed by conducting prediction using either data from multiple sensors or by considering information extracted (multiple consecutive time samples) of the explanatory stream. The effectiveness of SMR is demonstrated on a sensor node sleeping coordination problem by reducing energy consumption by more than an order of magnitude with respect to the best previously published technique.**

Index Terms— **Modeling, Multisensor systems**

I. Introduction

Modeling is a problem that inherently permeates many tasks in both sensor networks and computational sensing. Intersensor modeling aims to predict a reading at a particular sensor at a particular time using one or more readings taken from any sensors, including itself, at any sampled time moment. Our objective is to develop a systematic non-parametric data-driven approach for intersensor modeling that leverages on insights about the physical nature of sensing signals and enables efficient and consistent treatment of optimization software used by sensor network applications.

Using Intel Berkeley dataset of 54 sensors of three modalities (temperature, humidity and light) during a period of three weeks, we concluded in many situations modeling a single sensor reading from the reading of another sensor at the same time moment is inaccurate. In addition, standard regression models solely model from sensor X to sensor Y, without considering the quality of prediction from Y to X. Without facilitating a symmetric prediction model for X to Y and Y to X, if the model is applied for determining a series of values between sensor X and sensor Y, the prediction values will be incorrectly produced, leading to sequences of values which eventually converge at the point where the two regression models intersect. Lastly we observed that two sensors are most often well correlated only if they are exposed to similar or identical sets of sources of excitation of the same modality. If either source increase/decreases its intensity the impact will be the same on both sensors. Due to these observations, we have developed a symmetric monotonic regression (SMR). We also propose a simple and fast lower bound regression (LBR) approach for evaluating the achievable accuracy of SMR.

The SMR technique leverages on two mechanisms: symmetry and monotonicity. Symmetry ensures that one simultaneously takes into account the errors of predicting sensor A from sensor B and sensor B from sensor A. The main benefit of enforcing symmetry is that it enables consistent circular predictions. For example, if we predict sensor A from sensor B and consequently predict from the obtained value sensor A using symmetric regression, we will obtained exactly the starting value of sensor A. Note that until now no monotonic regression was providing this guarantee. Monotonicity is a constraint that enforces that for a pair of measurements at sensor A, m'_A and m''_A, such that m'_A is smaller than m''_A readings at sensors B in the corresponding moments m'_B and m''_B satisfy the condition that $m''_B \geq m'_B$. The intuition behind monotonicity is the following. Two sensors are well correlated most often only if they are exposed to similar or identical sets of sources of excitation of the same modality. If either source increase/decreases its intensity the impact will be the same on both sensors.

We first introduce SMR, LBR, and efficient polynomial-time optimal algorithms for their calculation. We show that SMR outperforms other popular regression and has accuracy that is close to one provided by LBR. Using SMR as an accurate prediction tool for inter-sensor modeling, we address the coordinated sleeping-based power management problem. SMR enables accurate inter-sensor modeling enabling minimal sampling at each sensor, which allows for efficient power optimization. The goal is to determine a set of sensors which can be alternately placed into a low-power sleep mode in a such a way that their missing sensor samples can be recovered using SMR prediction from awake and sampled sensors. The problem is optimally solved using integer linear programming (ILP) formulation. Another ILP formulation is used to solve 2-D SMR problem that is used for time-shifted prediction using data from two sensors.

II. Related Work

We briefly survey the most directly related work in this section. Nowak [1] presented an application of a distributed expectation maximization (EM) algorithm for density estimation in sensor networks. Coates [2] proposed distributed estimation of the current state at multiple sensor nodes via particle filtering. Delouille et al. [3] developed a new iterative distributed

algorithm for linear minimum mean-squared error (LMMSE) estimation in sensor networks whose measurements follow a Gaussian hidden Markov graphical model with cycles. In [4], a kernel linear regression approach for in-network modeling is presented. Their goal is to develop linear relationships between correlated sensors in such a way that constraints on the model parameters are communicated instead of the data itself. Paskin et al. [5] developed an architecture for distributed inference in a sensor network using a combination of graphical models and junction trees.

Pool Adjacent Violators Algorithm (PAV), proposed by Brunk in 1955 [6] was the first monotonic regression approach. One popular form of kernel smoothing after applying PAV is proposed by Mukerjee [7], who used the Nadaraya-Watson estimate, for this task. Dette and Pilz [8], proposed a technique that combines density with a regression estimate to obtain a monotone estimate of the inverse regression function. In 1964, Kruskal published a paper that in context of his multidimensional scaling solved the monotonic regression problem [9]. The approach is essentially identical to the PAV algorithm. It has been used and refined by a number of researchers in particular in the signal processing community [10], [11]. Koushanfar et al. [12] have developed a combinatorial isotonic regression (CIR) approach for intersensor modeling. We significantly improved the accuracy of this technique by simultaneously addressing both monotonicity (aka isotonicity) and symmetry.

One of the key issues in wireless sensor networks is power conservation. A number of techniques have been proposed at all levels of the design process from communication protocols [13] to digital signal processing [14]. Willet et al. introduced backcasting where adaptive sampling is applied for efficient field estimation [15]. Jain et al. [16] proposed an adaptive sampling approach which varies the sampling rate at each sensor and therefore adapting to the streaming-data characteristics of the sensor. The use of mobile sensor nodes are used to determine sampling density required in various environmental regions in [17]. Their Fidelity Driven Sampling actively seeks to minimize error without prior knowledge of the variable field.

Kar et al. [18] advocated and theoretically analyzed a technique for dynamic node activation in networks of rechargeable sensors. Koushanfar et al. [12] have developed an ILP formulation that addresses sleeping assuming only synchronous prediction of samples at one node from samples of an another single node. Their work can be considered as a very specialized case of our effort both in terms of modeling, number of techniques used to enable efficient sleeping, and a much simpler ILP formulation that is applicable only on smaller instances of the networks. All of the proposed sampling schemes assumed simultaneous sampling at all nodes. Our goal is to demonstrate that by relaxing this requirement and using time-shifted data for data recovery we can improve the lifetime of the network by more than an order of magnitude while maintaining the user specified level of accuracy.

III. SMR AND LOWER BOUND REGRESSION

In this section we present the symmetric monotonic regression algorithm, the lower bound regression measure, and

```
Procedure SMR (s_i, s_j)
    Initialize count matrix Γ = 0
    For t=1 to max_t, Γ(s_i, s_j) = Γ(s_i, s_j) + 1
    Γ_{s_i→s_j} = 1− normalized Γ by column sums
    Γ_{s_j→s_i} = 1− normalized Γ by row sums
    Initialize Graph G
    For y = 1 to number of rows of Γ - 1
        For x = 1 to number of columns of Γ - 1 {
            Create edge e_h from (x, y) to (x + 1, y)
            e_h.weight = Γ_{s_i→s_j}(x + 1, y)
            G.addedge(e_h)
            Create edge e_d from (x, y) to (x + 1, y + 1)
            e_d.weight = Γ_{s_i→s_j}(x+1, y+1)+Γ_{s_j→s_i}(x+1, y+1)
            G.addedge(e_d)
            Create edge e_v from (x, y) to (x, y + 1)
            e_v.weight = Γ_{s_i→s_j}(x, y + 1)
            G.addedge(e_v)
        }
    SMR = G.Dijkstra_Shortest_Path(0, 0, max(s_i), max(s_j))
```

Fig. 1. Basic SMR algorithm pseudo code.

propose a integer linear programming formulation for creation of a 2-D symmetric monotonic regression model.

Given the sensor measurements from two sensors s_i and s_j, the goal of SMR is the creation of a prediction model for $s_i \rightarrow s_j : \Psi(s_i)$ and $s_j \rightarrow s_i : \Psi'(s_j)$ which is symmetric and monotonic and minimizes the error in prediction from $s_i \rightarrow s_j$ and $s_j \rightarrow s_i$ simultaneously.

To begin a normalized density map of all sensor sample pairs (s_i, s_j) is created. Note, that while the binning of data can introduce additional error, we minimize the error by placing the bin size equivalent to the precision of the sensor readings. An error metric corresponding to each bin of the matrix is then calculated. This error is the amount of error introduced if the bin were to be selected as the predictor from one sensor to the other. The SMR problem is now stated as: find a mapping from s_i to s_j and from s_j to s_i in this matrix that follows the symmetric and monotonic restrictions while reducing the mapping errors. We use a dynamic programming approach to solve for this mapping. The pseudo code of the algorithm is listed in Figure 1.

The approach begins by creating a matrix Γ with dimensions corresponding to $|max(s_i) - min(s_i) \times max(s_j) - min(s_j)|$. For simplicity, we omit the details of the binning and the mapping of the raw data to each bin in Γ. Throughout the remaining discussions, we use the notation $\Gamma(s_i, s_j)$ to refer to the matrix bin corresponding to observed data pair (s_i, s_j) with s_j mapped to the row number, and s_i mapped to the column number in Γ. The binning resolution of Γ determines the granularity of the prediction function we compute.

Once the matrix Γ is created and initialized to 0, for all observed data pairs (s_i, s_j), we increment the element $\Gamma(s_i, s_j)$. From Γ, we compute two normalized matrices $\Gamma_{s_i \rightarrow s_j}$ and $\Gamma_{s_j \rightarrow s_i}$ by dividing each non-zero entry of Γ by column sums or row sums respectively and subtracting from 1. If an element $\Gamma_{s_i, s_j} = 0$, we set the corresponding element in $\Gamma_{s_i \rightarrow s_j}$ and $\Gamma_{s_j \rightarrow s_i}$ to 1.

The next step is building a graph G with nodes corresponding to the elements of Γ, and edges corresponding to monotonic movements from $(min(s_i), min(s_j))$ to $(max(s_i), max(s_j))$. Figure 2 (left) shows a simple version of such a

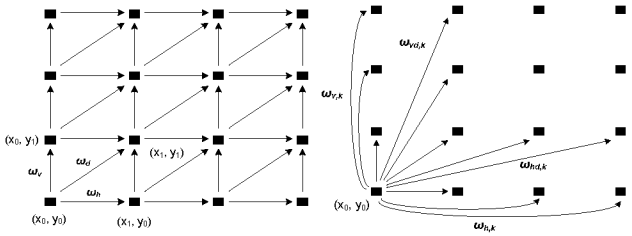

Fig. 2. Left: Basic monotonic search graph for SMR. Right: Monotonic SMR graph with "up-to-k"-jump edges.

graph with horizontal, vertical, and diagonal edges. Depending on the type of edge, starting in node (x_i, y_j) and ending in (x_e, y_e), we have the horizontal, diagonal, and vertical weights (ω_h, ω_d, and ω_v) that are computed as follows:

$$\omega_h = \Gamma_{s_i \to s_j}(x_e, y_e) \quad (1)$$

$$\omega_d = \Gamma_{s_i \to s_j}(x_e, y_e) + \Gamma_{s_j \to s_i}(x_e, y_e) \quad (2)$$

$$\omega_v = \Gamma_{s_j \to s_i}(x_e, y_e) \quad (3)$$

The basic monotonic graph G can be extended to allow "up-to-k"-jumps where k is a positive integer. We use the notation G_k to refer to this graph. Note that the basic graph described above is a special case of the extended graph with $k = 1$ (G_1). As shown in figure 2 (right), when constructing the G_k graph, for each node (x_i, y_j) we introduce the following edges:

- k horizontal edges to nodes $(x_{i+1}, y_j)...(x_{i+k}, y_j)$ with weight $\omega_{h,l}$
- k vertical edges to nodes $(x_i, y_{j+1})...(x_i, y_{j+k})$ with weight $\omega_{v,l}$
- k diagonal edges to nodes $(x_{i+1}, y_{j+1})...(x_{i+k}, y_{j+1})$ with weight $\omega_{hd,l}$
- $k-1$ diagonal edges to nodes $(x_{i+1}, y_{j+2})...(x_{i+1}, y_{j+k})$ with weight $\omega_{vd,l}$

The length of the edge is represented by l and can range from $1...k$. The error weights on each edge is computed as:

$$\omega_{h,l} = l \cdot min\{\Gamma_{s_i \to s_j}(x_{i+1}, y_j)...\Gamma_{s_i \to s_j}(x_{i+l}, y_j)\} \quad (4)$$

$$\omega_{v,l} = l \cdot min\{\Gamma_{s_j \to s_i}(x_i, y_{j+1})...\Gamma_{s_j \to s_i}(x_i, y_{j+l})\} \quad (5)$$

$$\omega_{hd,l} = l \cdot min\{\Gamma_{s_i \to s_j}(x_{i+1}, y_j)...\Gamma_{s_i \to s_j}(x_{i+l}, y_j)\} + \Gamma_{s_j \to s_i}(x_{i+l}, y_{j+1}) \quad (6)$$

$$\omega_{vd,l} = l \cdot min\{\Gamma_{s_j \to s_i}(x_i, y_{j+1})...\Gamma_{s_j \to s_i}(x_i, y_{j+l})\} + \Gamma_{s_i \to s_j}(x_{i+1}, y_{j+l}) \quad (7)$$

Once the search graph G is constructed, we use a dynamic programming approach based on Dijkstra's Shortest Path algorithm to compute the lowest cost path P_{opt} from the smallest to the largest pairs (corners) of Γ. The prediction functions Ψ and Ψ' are the mappings from $s_i \to s_j$ and $s_j \to s_i$ respectively, produced by this path P_{opt}.

The SMR algorithm complexity is quadratic with respect to the bin size selected for counting data pair occurrences.

A. LBR Algorithm

We have propose a lower bound regression (LBR) measure in order to evaluate the quality of the SMR technique. Small percentage error between the two regressions indicate that the evaluated regression technique is performing well. In addition, LBR can be used in order to determine if any inter-sensor model will predict with accuracy satisfactory to the user. The LBR can be easily calculated in linear time in terms of the available number of samples for any norm, such as L_1, L_2, and L_∞. In order to calculate the optimal LBR with respect to the L_1 norm, find for each value of variable B the median of corresponding values at variable A. Or, for L_2-optimal LBR to find the average value of values of variable A that occur for each value of variable B.

In order to illustrate the LBR measure and the close match between the prediction error between the LBR and the SMR technique, Figure 3 shows the prediction error of these two techniques for two sensor 6 and 7 from the Berkeley dataset for the three modalities: temperature, humidity, and light. Evaluation is conducted from a dataset that consists of eight days of sensor measurements. Note, that the LBR is derived using exactly this dataset while the SMR is developed using a learning dataset that consists of three days of measurements. As we see at the bottom of the figures the percentage difference between error of SMR and LBR are 0.41%, 34.9%, and 21.4% for temperature, humidity, and light respectively.

B. 2-Dimension Symmetric Monotonic Regression

When building the symmetric monotonic prediction model for two sensors it is assumed that the training data consists of (x, y, z) triples where each component is a sensor reading for the corresponding sensor. Given three sensors, x, y, and z, the goal is to determine the value for sensor z with minimal L_1 error, given the values of sensors x and y.

The symmetric monotonic prediction model can be addressed using an ILP formulation. Prior to formulating the problem, a preprocessing step is performed where the total L_1 error in the X and Y-axis is calculated for each possible prediction value of z at (x, y). By considering the error in both the X and Y directions, we are ensuring symmetry. We denote the preprocessed constant error values by E_{xyz}. The formulation contains one set of variables v_{xyz} which denotes if value z is selected as the SMR prediction model value for point the corresponding (x, y) position.

$$E_{xyz} - \text{total error in X and Y-axis at position } (x, y, z)$$

$$v_{xyz} = \begin{cases} 1, & \text{if value } z \text{ selected at } (x, y) \text{ position} \\ 0, & \text{otherwise.} \end{cases}$$

$$\text{for all } x, y : \sum_z v_{xyz} = 1 \quad (8)$$

$$\text{for all } x, y : \sum_z z v_{xyz} \geq \sum_z z v_{(x-1)yz} \quad (9)$$

$$\text{for all } x, y : \sum_z z v_{xyz} \geq \sum_z z v_{x(y-1)z} \quad (10)$$

$$Y = MIN(\sum_x \sum_y \sum_z E_{xyz} v_{xyz}) \quad (11)$$

There are three types of constraints in the ILP formulation of the problem. The first set of constraints is that for each (x, y) position only a single value for z may be selected. Eq.

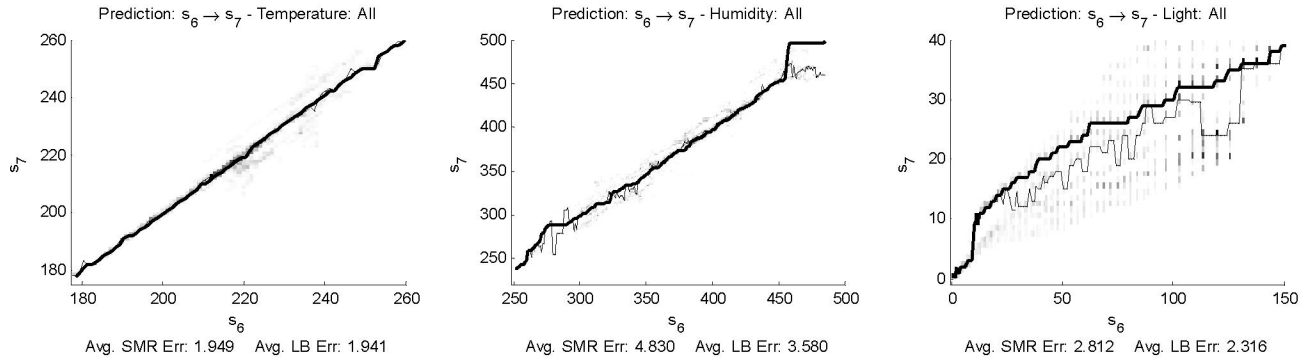

Fig. 3. SMR and lower bound models for sensors 6 and 7 for three modality over 8 days.

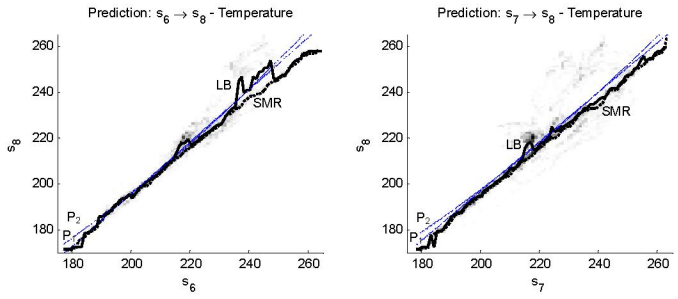

Fig. 4. SMR Model for prediction of sensor 8 from sensor 6 and 7 individually.

(8) denotes this constraint. In order to ensure that the model is monotonic to both the X and Y-axis, two sets of constraints are added. The first ensures that for each value selected for z in terms of the X-axis, that the value is always increasing (Eq. (9)). The constraint is formed by multiplying the constant value for the selected z position at the current position (x, y) by the selected position, and ensuring that this value is greater than the value of selected for the position $(x-1, y)$. An identical set of constraints is created to enforce the same condition in the Y-axis (Eq. (10)). Finally, Eq. (11) states the objective function for the problem which is to minimize the sum of all error in the X and Y-axis for each selected z value.

Figure 5 shows a plot of the 2-D SMR model for predicting temperature readings of sensor 8 from sensors 6 and 7. In Figure 4 the SMR models for prediction of sensor 8 using data from sensor 6 alone and using data from sensor 7 alone are shown. We see significant improvement (2.96 and 4.05 individually, 1.51 combined) in accuracy. In Table I we present the SMR, LBR, linear regression (P_1) and a quadratic fit (P_2) for each of the predictions individually for all three modalities (temperature, humidity, light) for sensor 6 or 7 predicting sensor 8. The average absolute SMR and LBR errors for 2-D prediction of sensor 8 from both sensor 6 and 7 was 1.51, 0.78 for temperature and 1.92, 0.91 for humidity, respectively. In this case, while their was a large improvement for combined prediction for temperature and humidity, for light this was not the case. The error in the LBR for light was 14.85 (no improvement over LBR of individual predictions), signifying that 2-D SMR would not perform better than the individual predictions.

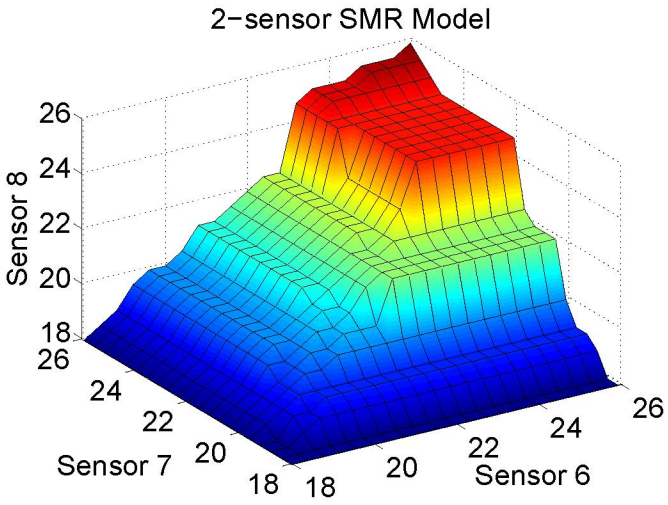

Fig. 5. 2-D SMR Model for prediction of sensor 8 from sensor 6 and 7 .

TABLE I
AVERAGE ABSOLUTE ERROR IN SMR, LBR, AND COMPARISON TO LINEAR REGRESSION (P_1) AND A QUADRATIC FIT (P_2).

Pair	$s_i \rightarrow s_j$				$s_j \rightarrow s_i$			
$i-j$	SMR	LBR	P_1	P_2	SMR	LBR	P_1	P_2
T6-8	2.96	2.60	3.28	3.23	2.96	2.24	2.93	3.32
T7-8	4.05	3.70	4.49	4.70	3.87	3.29	4.14	4.48
H6-8	6.38	5.64	7.83	9.46	5.95	4.78	7.15	6.45
H7-8	10.35	6.91	9.51	10.92	7.23	6.27	9.30	8.55
L6-8	9.69	5.57	12.03	21.5	6.61	5.12	6.49	8.61
L7-8	17.1	10.1	27.6	28.8	2.76	1.97	4.21	4.54

IV. EXPERIMENTAL RESULTS

In our experimental evaluation of the SMR models, we utilized the models to address the sleeping coordination problem. Models for sensor readings of temperature, humidity, and light taken from the Intel Berkeley dataset [19] we developed. The dataset consists of 55 sensors which were sampled at 30 second intervals. All models were built with three days of data and evaluated using eight different days. The sleeping coordination problem aims at maximizing the amount of time each sensor is placed into a minimal energy state, or sleep state. Any sensor that is not collecting sensor readings or communicating can be placed in this state and does not communicate. The goal is to minimize the number of samples

taken by each sensor, s.t. at every time step a sensor is sampled or its data can be predicted by another sensor at the same time step, by a previously sampled sensor, or by two previously sampled sensors. We model the problem as a special case of the Domatic Number Problem and is therefore NP-complete.

In the initial observations of the dataset we noted accurate prediction from sensor readings shifted in time. For this reason, evaluation of the sleeping coordination problem will be performed under the assumption of two additional types of SMR models: time-shifted SMR and 2-D SMR.

A. Sleeping Coordination ILP

The formulation for the problem assumes a periodic sampling window size W is prespecified. The goal is to assign each sensor to collect readings at the minimal number of epochs $t = \{0, \ldots, W\}$ in the window W. Variable s_{it} is used to denote whether sensor i is to be sampled at epoch t. The second set of variables, p_{it} is used to determine if a reading at epoch t for sensor i is predictable using any of the models. Finally, the last variable l is used to determine the maximum number of samples take by any single sensor in the window. The objective function is to minimize this value (Eq. (12)). Note that uppercase letters are used to refer to constant values, while lowercase letters refer to variables in the ILP formulation. Additionally, for the model variations we will use the capital letters D, and E. D to the SMR model built using time-shifted prediction values, and E for the 2-D SMR model.

$$W \quad - \quad \text{size of periodicity window}$$

$$s_{it} = \begin{cases} 1, & \text{if sensor } i \text{ is sampled at epoch } t \\ 0, & \text{otherwise.} \end{cases}$$

$$p_{it} = \begin{cases} 1, & \text{if sensor } i \text{ is predictable at epoch } t \\ 0, & \text{otherwise.} \end{cases}$$

$$l = \text{largest number of samples by any sensor}$$

$$d_{it}, e_{it} = \begin{cases} 1, & \text{if sensor } i \text{ is predictable by model} \\ & D, \text{ or } E \text{ at epoch } t, \text{ respectively} \\ 0, & \text{otherwise.} \end{cases}$$

$$q_{ij} = \begin{cases} 1, & \text{if sensor } i \text{ is simultaneously} \\ & \text{sampled with sensor } j \\ 0, & \text{otherwise.} \end{cases}$$

The ILP formulation includes three sets of constraints. The first set, Eq. (13), specifies that for each sensor the sum of the samples taken by that sample must be less than or equal to l, the largest number of samples taken by any sensor. The second set of constraints (Eq. (14)) specifies that at each epoch, each sensor must be either sampled or/and predicted. Finally, the constraint in Eq. (15) specifies that if a sensor i is to be predicted at epoch t, then the it must be predicted by one of the models (D (TSMR), or E (2-D SMR)).

$$Y = MIN(l) \tag{12}$$

$$\text{for all } i : \sum_t s_{it} \leq l \tag{13}$$

$$\text{for all } i, t : s_{it} + p_{it} \geq 1 \tag{14}$$

$$\text{for all } i, t : d_{it} + e_{it} \geq p_{it} \tag{15}$$

It is important to note that each model may not be possible or necessary for prediction of a particular sensor. Therefore Eq. (15) must be modified only to include the feasible models for predicting the particular sensor i at each epoch t. We introduce variables d_{it}, and e_{it} in order to formulate the prediction cases for sensor i at epoch t using any of the two models (TSMR, 2-D SMR), respectively.

The constraint for the time-shifted single sensor SMR model (D) ensures that a sensor i is predictable using model D at each epoch (ie. $d_{it} = 1$ if and only if at least one sensor j can predict sensor i within the time-shifted value D_{ij}). For each sensor i at each possible epoch t the predictability variable, d_{it}, is calculated. In the on-line case, which we are considering, the value is only predicted from previously sampled sensor readings. Therefore, for each possible predictor sensor j if j is sampled at any epoch between $t - D_{ij}$ and t then i is predictable at t. To formulate this constraint we calculate the sum of all sampled sensors j which occur within the time period τ=$(t-D_{ij}), \ldots, t$. If there are no sensor readings which can be used to predict sensor i at epoch t (ie. summation is zero), then d_{it} must be assigned to zero. However, if the summation is one or more, then d_{it} can be assigned to zero or one. This is acceptable because Eq. (14) will ensure that sensor i is sampled at time k or that it is predictable, and therefore forcing d_{ik} to one if necessary.

Note that in Eq. (16) we denote the calculation of the time period for predictability of each sensor using modulus W. Since W is the duration of the periodicity window, it is acceptable for a sensor to be predicted from a previous window under the assumption that the epoch it is within the specified time-shift. Therefore, the modulus of time position of the sampled sensor reading is taken into account for the time-shift.

$$D_{ij} \quad - \quad \text{max time-shift for sensor } j \text{ to predict } i$$

$$E_{ijk} \quad - \quad \text{prediction ability of } i \text{ from sensor } j \text{ and } k$$

$$\text{for all } i, t : \sum_j \sum_{\tau=0}^{D_{ij}} x_{j[(t-\tau)\%W]} \geq d_{it} \tag{16}$$

$$\text{for all } i, j, k, t, |E_{ijk}| > 0 : s_{j[(t-v_j)\%W]} + s_{k[(t-v_k)\%W]} \geq e_{it} \tag{17}$$

The final constraint is for addressing the use of the 2-D SMR model (E). If two sensors j and k can predict sensor i above $P\%$ accuracy, then i is predictable at t. Note that the constraint in Eq. (17) is to be written for any time shifted combination of epochs which are the time phased pairs in the set E_{ijk}.

B. Analysis

Our analysis was performed on 2 types of ILP instances: single sensor SMR prediction (the standard base case), single sensor time-shifted SMR prediction, and 2-D time-shifted SMR prediction. The single sensor case is when each sample can only be predicted from other single sensor samples taken at the same time moment. In the single sensor time-shifted case each sample can be predicted from other single sensor samples taken within the maximum time-shift allowed by the pre-specified prediction error value (TSMR). In the final case, the addition of two time-shifted sensor readings are incorporated for prediction (2-D SMR).

Each case is formulated using varied input to an ILP formulation. All formulations were solved with CPLEX with a maximum runtime of 5 minutes, but almost all instances ran within seconds. Three sets of L_1 errors were considered 2%, 3%, and 5%. In addition, three instances of the dataset were examined: all sensor nodes, a set of approximately two-thirds of the nodes, and a set of nodes from one third of the layout area.

In Table II we present the node assignment results for temperature, humidity, and light. In the first column we show the number of nodes, followed by the maximal L_1 prediction error allowed in prediction. Then for temperature and humidity, we shows the improvement for the single sensor simultaneous prediction over no optimization, followed by single time-shifted sensor prediction. The final column for the modality shows the improvement of the 2-D SMR prediction models. We see that as the amount of error allowable is increasing, the savings decreases. This is due to the fact that with increased model prediction error a higher number of sensors can predict other sensors over long time-shifts. The final column of the table shows the results for light using the 2-D SMR models along with the SMR and TSMR models. For light this case is the only feasible case since the error in prediction using a single sensor higher than allowed.

The most important result is that even for very low error, 2%, the node assignment approach is capable of performing 20 times better than the standard base case sleeping strategy, translating into 20 times longer lifetimes for the network with the single sensor prediction and two orders of magnitude better prediction when considering 2-D sensor prediction models. For humidity we see the same patterns, occurring as we increase the allowable error. Nevertheless, the node assignment approach still was able to achieve 22-120 times more energy savings over the base case. Finally, even for light we were able to achieve a 2-19 times improvement.

TABLE II
EXPERIMENTAL RESULTS FOR INTEL BERKELEY DATASET.

		Temperature			Humidity			Light
#	L_1 Err	SMR	TIME SMR	2-D SMR	SMR	TIME SMR	2D SMR	2-D SMR
51	2%	1	21	41	2	22	84	2
34	2%	2	34	70	2	22	90	2
17	2%	6	120	240	4	44	160	4
51	3%	2	44	260	2	22	240	4
33	3%	4	44	280	4	34	320	4
17	3%	8	42	340	4	34	340	6
51	5%	24	60	150	10	60	140	19
34	5%	40	200	400	10	120	420	17
17	5%	16	160	340	4	44	340	16

V. CONCLUSION

We have presented a symmetric monotonic regression (SMR) technique for intersensor modeling and its application to energy efficient data collection in sensor networks. We have developed an approach for calculating the lower bounds of prediction errors and use it for analysis of the quality of SMR. We illustrate the application of the SMR technique to the sleeping coordination problem for power optimization in sensor networks in three forms symmetric monotonic regression for prediction from a single sensor, using time-shifted sensor readings, and through a pair of sensors. The lifetime of the sensor network is shown to improve by more than an order of magnitude in comparison with the best previously obtained results.

REFERENCES

[1] R. Nowak, "Distributed EM algorithms for density estimation and clustering in sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2245–2253, 2003.

[2] M. Coates, "Distributed particle filters for sensor networks," in *IPSN*, 2004, pp. 99–107.

[3] V. Delouille, R. Neelamani, and R. Baraniuk, "Robust distributed estimation in sensor networks using the embedded polygons algorithm," in *IPSN*, 2004, pp. 405–413.

[4] C. Guestrin, P. Bodi, R. Thibau, M. Paski, and S. Madde, "Distributed regression: an efficient framework for modeling sensor network data," in *IPSN*, 2004, pp. 1–10.

[5] M. Paskin, C. Guestrin, and J. McFadden, "A robust architecture for distributed inference in sensor networks," in *IPSN*, 2005, pp. 55–62.

[6] H. Brunk, "Maximum likelihood estimates of monotone parameters," *Ann. of Math. Statistics*, vol. 26, pp. 607–616, 1955.

[7] H. Mukerjee, "Monotone nonparametric regression," *Ann. of Statistics*, vol. 16, pp. 741–750, 1988.

[8] H. Dette and K. Pilz, "A comparative study of monotone non-parametric kernel estimates," Tech. Rep. 21, Universitt Dortmund, Sonderforschungsbereich, 2004.

[9] J. Kruskal, "Nonmetric multidimensional scaling: a numerical method," *Psychometrika*, vol. 29, pp. 115–129, 1964.

[10] Q. Stout, "Optimal algorithms for unimodal regression," *Computing Science and Statistics*, vol. 32, 2000.

[11] N. Sidiropoulos and R. Bro, "Math. programming algorithms for regression-based nonlinear filtering in IR," *IEEE Trans. on Signal Processing*, vol. 47, no. 3, pp. 771–782, 1999.

[12] F. Koushanfar, N. Taft, and M. Potkonjak, "Sleeping coordination for comprehensive sensing using isotonic regression and domatic partitions," in *Infocom*, 2006.

[13] W. Ye, and and others., "An energy-efficient MAC protocol for wireless sensor networks," in *Infocom*, June 2002, pp. 1567–1576.

[14] A. Wang and A. Chandrakasan, "Energy-efficient DSPs for wireless sensor networks," *IEEE Sig Proc Magazine*, vol. 43, no. 5, pp. 68–78, 2002.

[15] R. Willett, and others., "Backcasting: adaptive sampling for sensor networks," in *IPSN*, 2004, pp. 124–133.

[16] A. Jain and E. Y. Chang, "Adaptive sampling for sensor networks," in *DMSN*, 2004, pp. 10–16.

[17] M. A. Batalin, and others., "Call and response: Experiments in sampling the environment," in *Sensys*, 2004, pp. 25–38.

[18] K. Kar, A. Krishnamurthy, and N. Jaggi, "Dynamic node activation in networks of rechargeable sensors," in *Infocom*, 2005.

[19] M. Paskin, and others., "A robust architecture for distributed inference in sensor networks," in *IPSN*, 2005.