Active Inference and Dynamic Gaussian Bayesian Networks for Battery Optimization in Wireless Sensor Networks

Caner Komurlu and **Mustafa Bilgic**
Illinois Institute of Technology
10 W 31st Street, Stuart Building
Chicago, Illinois 60616
ckomurlu@hawk.iit.edu, mbilgic@iit.edu

Abstract

Wireless sensor networks play a major role in smart grids and smart buildings. They are not just used for sensing, but they are also used as actuating. In terms of sensing they are used to measure temperature, humidity, light, to detect motion, etc. Sensors are often operated on a battery and hence we often face a trade-off between obtaining frequent sensor readings versus maximizing their battery life. There have been several approaches to maximizing their battery life from hardware level to software level such as reducing components energy consumption, limiting node operation capabilities, using power-aware routing protocols, and adding solar energy support. In this paper, we introduce a novel approach: we model the sensor readings in a wireless network using a dynamic Gaussian Bayesian network (dGBn) whose structure is automatically learned from data. dGBn allows us to integrate information across sensors and infer missing readings more accurately. Through active inference for dGBns, we are able to actively choose which sensors should be pulled for a reading and which ones can stay in a power-saving mode at each time step, maximizing prediction accuracy while staying within the budgetary constraints on battery consumption.

Introduction

Smart buildings are increasingly more prevalent and they are used in various service areas such as factories, energy facilities, airports, harbors, schools, medical centers, government buildings, military bases, etc. Properties of smart buildings and their objectives show a wide range of variety: security, surveillance, adapting to weather conditions and quality of service.

Sensing is a major task in smart buildings. Arguably the most common sensing mechanism consists of wireless sensor networks (WSN). WSNs' key components are sensor nodes. They are mechanically independent and they communicate through RF. They sense one or multiple specific physical events, process their readings, and forward to a server.

As nodes in a WSN are mechanically independent, they rely on their individual source of energy, most generally a battery. Lifespan of each sensor node is determined by how fast it consumes energy. Many studies have been carried out to increase lifespan of nodes in WSN. In addition to extending battery life through physiochemical enhancements, some strategies have been proposed for decreasing energy expenditure (Akyildiz et al. 2002).

A sensor node performs three essential tasks: sensing in which it converts a physical quantity to a reading, processing in which it treats readings and store them, and finally communicating in which it sends or receives data. Energy consumption of sensing and processing is negligible beside communicating. Instead of gathering every reading from each node, readings from a subset can be fetched and remainders can be kept silent. The more frequent a sensor keeps silent, the longer it survives.

The intuition behind keeping some sensors silent is to utilize the correlations between sensors to predict sensor readings that are not communicated to conserve energy. Collecting readings from a sensor or a group of sensors might give enough data to estimate readings of another. Thereby, we can save energy by not fetching true readings. Correlations existing between sensors should be considered not only in space but in time as well. A sensor's reading in the past might provide enough insight to estimate its value for some period with help of concurrent readings from other sensors and their past readings as well.

In this paper, we propose a dynamic Gaussian Bayesian network (dGBn) model for WSNs and utilize the correlation between sensors to minimize prediction error subject to how many messages each sensor can send in a given time frame. Our contribution consists of two phases. First we model a WSN as a dGBn that represents correlations between sensors in space and time. Second, on top of our dGBn model, we select sensors to fetch readings on each time stamp with an intelligent strategy, which we call *active inference*. Active inference dynamically selects random variables on dGBn, which are sensors in this context, for observation. This observation is used as evidence to maximize accuracy. As a result, active inference can help us to minimize battery usage.

In the rest of the paper, we provide background on wireless sensor networks and dynamic Bayesian network models. Then we describe our dynamic Gaussian Bayesian network model and we formulate the active inference problem followed by a discussion of experimental methodology. We finally present results, discuss related work and conclude.

Background and Approach

Problem Statement

Lifespans of sensors in a typical wireless sensor network (WSN) depend on their battery consumption and communication is the largest consumer of battery. To increase lifespan, an obvious solution is to reduce the frequency of messages that a sensor sends to other sensors and/or to a central server. The downside of reducing the frequency of messages is that it leads to missing information at various time stamps. One possible remedy is to use a model of sensor readings and i) when the sensor reading is available, use the observed reading, and ii) when the sensor reading is missing due to reduced communication frequency, predict missing values using the underlying model.

In this paper, we tackle the following problem: *given a sensor network, a predictive model for the sensor network, and a budget of how many messages can be sent per time stamp, determine the ideal sensors that should communicate their readings to a central location so that the overall error over the network at each time stamp is minimized.* More formally, let Y_t^s be the sensor reading for time t and sensor s (if it is communicated to the central server, then it is observed; if not, it is predicted), $\mathcal{Y}_t$ be the sensor readings, both observed and predicted, of all sensors at time t (i.e, $\mathcal{Y}_t$ is the union of Y_t^s for all s), X_t^s be the always-observed features of the sensor s at time t, which includes sensor specific information such as ID and location, $\mathcal{X}_t$ (i.e, $\mathcal{X}_t$ is the union of X_t^s for all s), the set of observed features of all sensors at time t, B the budget, t the current time, $\mathcal{O}$ the the set of observed readings up to time t (i.e., the communicated Y_i^s values for $0 \leq i < t$), θ the underlying model, and Err the error function over $\mathcal{Y}$, the objective is:

$$\underset{\mathcal{S} \subset \mathcal{Y}_t}{\operatorname{argmin}} \operatorname{Err}(\mathcal{Y}_t | \mathcal{S}, \mathcal{O}, \mathcal{X}_t, \theta) \;\; s.t. \; |\mathcal{S}| = B \tag{1}$$

That is, find the subset of sensors at time t that should communicate their readings to the central location so that the error over all sensor readings, observed and predicted, is minimized for time t. Note that in this case, the error is computed over both observed and predicted sensors. The error for observed sensors can be assumed to be zero (or in the case of noisy sensor readings, the error can reflect the noise in readings). Alternatively, one can compute the error over only the sensors that did not communicate their readings and hence need to be predicted:

$$\underset{\mathcal{S} \subset \mathcal{Y}_t}{\operatorname{argmin}} \operatorname{Err}(\mathcal{Y}_t \setminus \mathcal{S} | \mathcal{S}, \mathcal{O}, \mathcal{X}_t, \theta) \;\; s.t. \; |\mathcal{S}| = B \tag{2}$$

We will discuss potential approaches to choosing $\mathcal{S}$ in active inference section below. There are a number of potential choices for θ. For example, one can train a temporal model per sensor such as a Gaussian process model per sensor. Another possibility is to use graphical models. We discuss these next.

Predictive Models for WSNs

Gaussian Processes One of the simplest and yet most appropriate strategy for modeling sensor readings over time is Gaussian Processes (Rasmussen 2004). For WSN, a possible approach is to train one Gaussian Process per sensor using its past history of readings over time. The advantage is that training of a Gaussian process per sensor and using it for prediction are both very fast. The disadvantages are, however, i) the correlations between sensors are not taken into account, and ii) once the model is trained and fixed, the observed readings at one time stamp at prediction time does not affect the predictions at other time stamps.

Dynamic Gaussian Bayesian Network We use dynamic Gaussian Bayesian network modeling to handle the temporal behavior of the system and to exploit spatio-temporal correlations between sensors. Each sensor is represented by a Gaussian random variable Y_t^s with conditional Gaussian distribution $\mathcal{N}(\beta_0 + \boldsymbol{\beta}^T \cdot \boldsymbol{Pa(Y_t^s)}, \sigma_i)$ where $\boldsymbol{Pa(Y_t^s)}$ is Y_t^s's parents. To explore independencies between sensors, thus finding a DBN structure capturing correlations the most, we resort K2 structure learning proposed by Cooper and Herskovits (1992).

To keep structure simplistic for temporal dependencies, we assumed that each random variable at time slice t, Y_t^s will be independent from all other sensor values of the previous time slice $t-1$ given itself in the previous time slice $t-1$ ($Y_t^s \perp \mathcal{Y}_{t-1} \setminus \{Y_{t-1}^s\} | Y_{t-1}^s$). Therefore we added Y_{t-1}^s to the parent list of Y_t^s

Since exact inference methods are intractable (Koller and Friedman 2009) on DBNs, approximate inference methods are generally resorted for prediction. We used Metropolis-Hastings (Hastings 1970) (MH) sampling for inference, an MCMC method which has found practice in many studies (Gilks, Richardson, and Spiegelhalter 1996). For each random variable, MH samples a value from a proposal distribution. Then using this newly sampled value and previously sampled value, it computes an acceptance probability. If the acceptance probability exceeds a random threshold sampled uniformly from $[0, 1]$ interval, then the new value is accepted as the new state of the random variable, otherwise the previous value is repeated in the sample.

Equation 3 shows how acceptance probability is computed and evaluated. $\mathrm{Y_t^s}$ is random variable Y_t^s's last accepted value, $\mathrm{Y_t'^s}$ is newly sampled value, $\boldsymbol{u^s}$ is Y_t^i's Markov blanket and r is random threshold for α the acceptance probability. Note that t represents here sampling step.

$$\begin{aligned}
&\alpha(\mathrm{Y_t'^s}, \mathrm{Y_t^s}, \boldsymbol{u^s}) = min\left\{1, \frac{p(\mathrm{Y_t'^s}, \boldsymbol{u^s}) \times q(\mathrm{Y_t'^s} | \mathrm{Y_t^s}, \boldsymbol{u^s})}{p(\mathrm{Y_t^s}, \boldsymbol{u^s}) \times q(\mathrm{Y_t^s} | \mathrm{Y_t'^s}, \boldsymbol{u^s})}\right\} \\
&r \sim U([0, 1]) \\
&\mathrm{Y_{t+1}'^s} = \begin{cases} \mathrm{Y_t'^s}, & \text{if } \alpha \geq 1 \\ \mathrm{Y_t^s}, & \text{otherwise} \end{cases}
\end{aligned} \tag{3}$$

Active Inference

Active inference (Bilgic and Getoor 2009), as a method in the context of statistical prediction such as classification or regression, is a technique of selective information gathering with the objective of maximizing prediction accuracy. Unlike active learning (Settles 2012) which collects label information during phase of training a statistical model, active inference collects in the phase of prediction.

Active inference on Bayesian networks is selecting random variables to observe during inference for better prediction on the rest. Active inference formulation is perfect match for battery optimization in WSNs, because active inference determines which nodes should be observed (i.e., which sensors should communicate their readings) to maximize the prediction accuracy on the remaining ones, subject to budgetary constraints.

Arguably, the simplest way is randomly selecting observation set. Another method is to define a fixed frequency for the sensors and hence they communicate their readings at fixed intervals, like once every hour. We call this approach, the *sliding window selection* approach. In this study we propose impact-based selection method, in which variables are selected for observation based on their impact on predicting others. In following subsections, we describe these three active inference methods.

Random Selection At each prediction time t, random selection chooses B random sensors for observation. Then, the observed readings for these sensors, and all the past observed readings are used during inference to predict the sensor readings for the unobserved ones and error is computed using Equations 1 and 2.

Sliding Window Selection This method, as a baseline alternative to random selection, selects sensors not randomly but with respect to a predefined order. In details, the list of sensors is shuffled initially and for once, then it is split into equal width bins, $b_1, b_2, \ldots b_m$. Then at each prediction time, a bin is selected in the original order, for observation. When we predict the time slice t_i, we observe the bin b_i. After mth time slice, we restart the sequence with b_1. To make picture clear, a very simple example can be given. In the case of 50% of sensors are selected as evidence. Some 25 sensors are selected for the first time slice, and the other 25 are observed in the following. On the third time slice, the first 25 are observed again, and so on. An advantage of this selection method is collecting information from each sensor in equal frequency, whereas random selection can choose a sensor multiple times in consecutive slices.

Impact-Based Selection For the impact-based selection approach, we first define the impact of each random variable on others when it is observed. Note that when a sensor's reading is observed, its immediate impact is on its unobserved neighbors in the network, i.e., its unobserved parents and unobserved children in the Bayesian network. In Equation 4 we define an impact function which incorporates $\boldsymbol{\beta}$ coefficients of Y_t^s and $\boldsymbol{\beta}$ coefficients of its children. ω_1 is an indicator function which is 1 when corresponding parent is unobserved, 0 otherwise. $\boldsymbol{\omega_1}$ is the vector of indicator functions of $\boldsymbol{Pa(Y_t^s)}$. $\boldsymbol{\omega_2}$ is the set of unobserved children of Y_t^s. The impact of observing Y_t^s on its immediate neighbors is defined as:

$$\mathrm{Imp}(x_i) = |\boldsymbol{\beta_i^T}| \cdot \boldsymbol{\omega_1} + \sum_{j \in \boldsymbol{\omega_2}} |\boldsymbol{\beta_j^{(i)}}| \qquad (4)$$

$$\underset{\mathcal{S}}{\mathrm{argmax}} \sum_{Y_t^s \in \mathcal{S} \subset \mathcal{Y}_t} \mathrm{Imp}(Y_t^s) \;\; s.t. \; |\mathcal{S}| = B \qquad (5)$$

Given this impact formulation, we would like to find a subset of random variables of which sum of impacts is maximum, as shown in Equation 5. In literature, this problem is also described as *finding the maximum cut given parts of size* in a graph. This problem is discussed by Ageev and Sviridenko (1999). They also show that it is *APX-Hard*, i.e. it does not admit an error guarantee with a polynomial-time approximation algorithm. Therefore, we propose a greedy algorithm.

Algorithm 1 Greedy Impact Based Selection

```
procedure GREEDY–IBS($\mathcal{Y}_t$,$B$)
    for $i = 1, 2, \ldots B$ do
        $maxI$ = 0
        $maxY$ = NIL
        for $Y_t^s \in \mathcal{Y}_t \setminus \mathcal{S}$ do
            if Imp($Y_t^s$) $> maxI$ then
                $maxI$ = Imp($Y_t^s$)
                $maxY$ = $Y_t^s$
            end if
        end for
        $\mathcal{S} = \mathcal{S} \cup \{maxY\}$
        for each $Y_t^s \in Pa(maxY)$ do
            $\omega_2(Y_t^s) = \omega_2(Y_t^s) \setminus \{maxY\}$
        end for
        for each $Y_t^s \in Child(maxY)$ do
            $\omega_1(Y_t^s)[maxY] = 0$
        end for
    end for
end procedure
```

Algorithm 1 describes our greedy impact based selection method. $\mathcal{Y}_t$ is the set of all random variables, B is the number of random variables we can select for observation and $\mathcal{S}$ is the observation/evidence set. We loop over all random variables which are not yet in the evidence set, we find the one with the highest impact, we add it to evidence list and remove it from parent list of its children and child list of its parents. The reason is obvious, a random variable has no impact on an observed variable, therefore once a variable is observed, it has to be removed from neighborhood of its neighbors and impact of all its neighbors should be recomputed. We repeat this selection B times.

Experimental Design and Evaluation

We first describe the data on which we test our dGBn model and active inference. Next we illustrate our baselines to assess our dGBn model and our active inference method.

Data

In our study we used Intel Research Lab sensor data (Deshpande et al. 2004). It consists of temperature, humidity, light, and voltage readings collected from 60 sensors. These sensors were placed in an office environment and employed for sensing the environment in different frequencies.

In this article, we focused on temperature readings for days 2, 3, 4, and 5 as these days show highest temperature variation. We split this period into equal bins of 30 minutes. We averaged readings of each sensor at each bin. In the end, we obtained a data set composed of one temperature value for each sensor at each time bin. We selected days 2, 3, 4 as training set, and we used first 6 hours of the 5th day as test set.

Evaluation Metrics

To evaluate our dGBn model, our active inference method, and the baselines, we utilize Mean Absolute Error (MAE), as our predictions are in a continuous domain. MAE measures distance of each reading to real value and computes mean of distances of a set of readings. Note that in the context of this study, a reading can be an actual observation, as well as a prediction. Based on our objective, we define the set of readings on which we compute the mean.

Mean Absolute Error on all readings In the first method, we refer to the objective function given in Equation 1. In this objective, we include all readings from the prediction time to compute error. Hence we average absolute error over all readings, including observed ones, in order to explore overall prediction performance of the predictive model (i.e. Gaussian processes or dGBn). Equation 6 shows how we compute average error of prediction over all sensors, given Err as the distance of a reading to its actual value:

$$MEA(t) = \frac{1}{|\mathcal{Y}_t|} \sum_{Y_t^s \in \mathcal{Y}_t} |\operatorname{Err}(Y_t^s)| \tag{6}$$

Mean Absolute Error on predicted readings In the second method, we refer to Equation 2 which minimizes error on only predicted readings. Equation 7 shows how we compute average error.

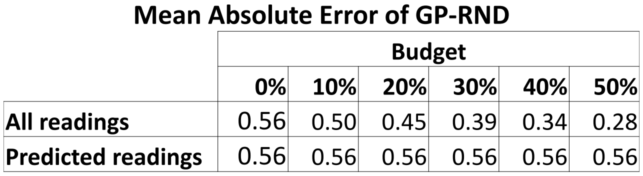
$$MEA(t) = \frac{1}{|\mathcal{Y}_t \setminus \mathcal{S}|} \sum_{Y_t^s \in \mathcal{Y}_t \setminus \mathcal{S}} |\operatorname{Err}(Y_t^s)| \tag{7}$$

For stochastic methods, which are random selection and sliding window selection, we ran 5 trials and averaged MAE at each time slice over 5 trials. Because a time slice has no specific effect on error reduction, we average MAE over time slices and obtain an overall error for each budget B.

In order to see the effect of budget on error reduction, we tried various budget levels: 0%, 10%, 20%, 30%, 40%, and 50%. To evaluate our dGBn model and our greedy impact-based selection method, we tried Gaussian processes with random selection, which we shortly call GP-RND. Then we tried our dGBn model with 3 different active inference methods: random selection (dGBn-RND), sliding window selection (dGBn-SW), greedy impact-based selection (dGBn-IBS).

Experimental Results and Discussion

We first present results for Gaussian processes followed by the results for dynamic Gaussian Bayesian network model. Then, we compare Gaussian processes with dynamic Gaussian Bayesian networks. We present these results in tables. In each table, columns are reserved for different budgets. The first row represents MAEs over all readings, whereas the second row shows MAEs over predicted readings only. For all cases, we present results corresponding to MAE over all readings and MAE over only predicted readings. We show these results in bar plots.

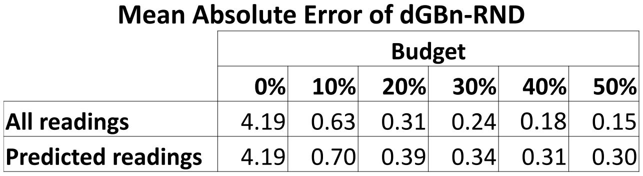


Mean Absolute Error of GP-RND

	Budget					
	0%	10%	20%	30%	40%	50%
All readings	0.56	0.50	0.45	0.39	0.34	0.28
Predicted readings	0.56	0.56	0.56	0.56	0.56	0.56

Table 1: Mean absolute error of random selection on Gaussian process with budgets from 0% to 50%.

Table 1 shows prediction error of Gaussian Process using random sampling as active inference (GP-RND). In the first row, we see a strictly monotonic reduction in error as budget increases. This result is not surprising: as the observed sensor percentage increases, the error goes down because the error on the observed ones are assumed to be zero. It is important to note however that as the observed sensor rate goes up, so does the battery consumption. On the other hand, in the second row, we notice that the error rate is invariant on budget for MAE with predicted readings only. Since Gaussian process cannot incorporate evidence and cannot make use of correlations between sensors, more evidence will not help predicting a reading.



Mean Absolute Error of dGBn-RND

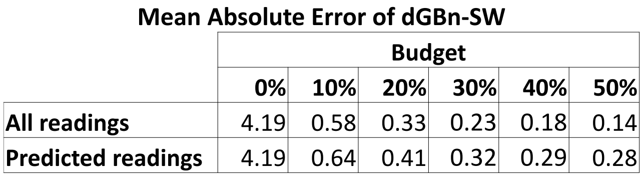
	Budget					
	0%	10%	20%	30%	40%	50%
All readings	4.19	0.63	0.31	0.24	0.18	0.15
Predicted readings	4.19	0.70	0.39	0.34	0.31	0.30

Table 2: Mean absolute error of random selection on dGBn with budgets from 0% to 50%.

In Table 2, we present performance result of dGBn with random selection. In both rows we see a monotonic decrease in error rates as more evidence is provided. The decrease in error in the first row is not surprising again, as was explained for the GP-RND result (Table 1). A decrease in error computed on predicted readings demonstrates that our dGBn model is able to exploit relationships between sensor nodes and hence observed/collected readings help reduce the prediction error for the unobserved/not-collected readings.

In Table 3 we show the effect of sliding window selection with dGBn (dGBn-SW) on error reduction. We see similar trends as dGBn-RND results. More evidence yields less error in regards to both error computation methods.

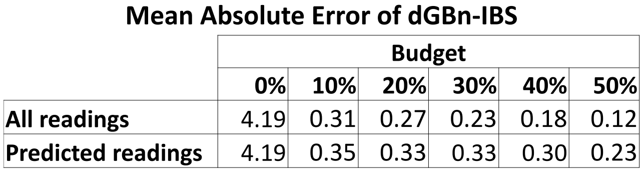
Table 4 shows our greedy impact-based selection method's contribution to our dGBn in error reduction. We see a monotonic decrease in both measures, except 30% on MAE averaged on unobserved sensors only, which is same as 20%.



Mean Absolute Error of dGBn-SW

	Budget					
	0%	10%	20%	30%	40%	50%
All readings	4.19	0.58	0.33	0.23	0.18	0.14
Predicted readings	4.19	0.64	0.41	0.32	0.29	0.28

Table 3: Mean absolute error of sliding window selection on dGBn with budgets from 0% to 50%.



Mean Absolute Error of dGBn-IBS

	Budget					
	0%	10%	20%	30%	40%	50%
All readings	4.19	0.31	0.27	0.23	0.18	0.12
Predicted readings	4.19	0.35	0.33	0.33	0.30	0.23

Table 4: Mean absolute error of impact-based selection on dGBn with budgets from 0% to 50%.

Next, we compare various modeling and active inference results side by side using bar plots. Figure 1 shows error rates of GP-RND, dGBn-RND, dGBn-SW, and dGBn-IBS, referring to all readings including observed ones. Likewise, Figure 2 shows error rates with respect to predicted readings only. In these plots, Y axis represents the error rate in temperature. X axis is reserved for evidence rates. We did not include error rates on budget 0% since they are drastically larger on this budget, and this difference scales the plot in a way that comparison of error rates on other budgets become infeasible.

In Figure 1, surprisingly, dGBn-RND and dGBn-SW performed worse than GP-RND on budget of 10%. We can interpret that this budget was good enough for neither RND nor SW on dGBn to reduce error as much as GP-RND can do with the same budget. Note that GP uses local attributes as well. However, on this rate we can see that dGBn-IBS outperforms all other model and method combinations including GP-RND. On other evidence rates, all active inference methods on dGBn outperform GP-RND. We can conclude that on overall prediction performance our dGBn model outperforms GP when at least 20% of sensors are observed. On 30% and 40%, dGBn-IBS outperforms dGBn-RND and it competes with dGBn-SW. When budget is 50%, dGBn-IBS is the best model and method combination.

Figure 2 presents error rates of each model and method combination with respect to predicted readings only. Results in this figure show some similarities to those on Figure 1. A major difference in Figure 2 is that GP-RND is constant with respect to budget. This is simply because GP-RND cannot exploit evidence to reduce error on predictions. On 30% and 40%, dGBn-IBS is at least as good as dGBn-SW, and better than dGBn-RND and GP-RND. On remaining rates, it is better than all other model and method combinations.

We omitted results for GP with SW approach because, given enough trials, the results are expected to be similar to GP-RND results. The reason is that GP per sensor approach does not exploit relationships between sensors.

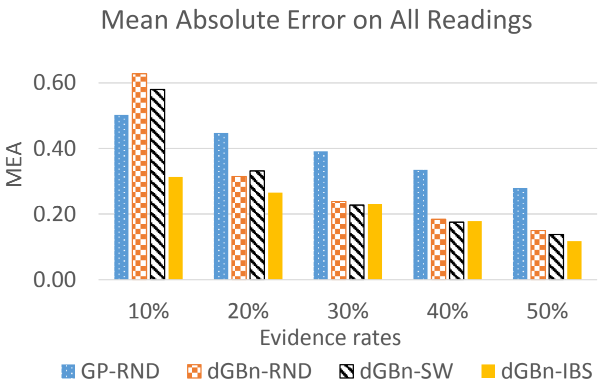


Figure 1: Mean absolute error of all model and method combinations computed over all readings on budgets from 0% to 50%

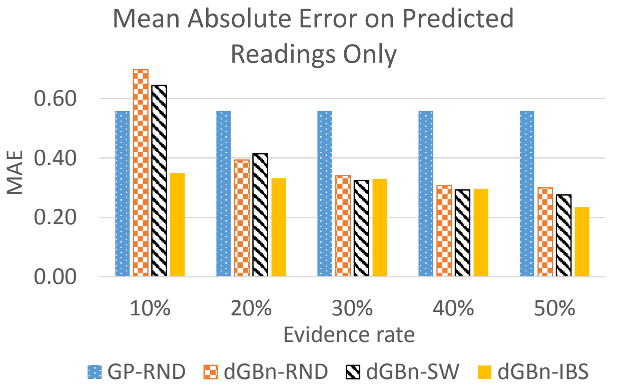


Figure 2: Mean absolute error of all model and method combinations computed over predicted readings only on budgets from 0% to 50%

Related Work

Many studies addressed predictive models for energy saving on wireless sensor networks (WSN). Elnahrawy and Nath (2004) propose prediction by context awareness. They discretize readings and use naïve Bayes classifier to predict these discrete values. They use the geographical topology of the sensor network as the basis of node neighborhood in their model.

In a similar way, Wang and Deshpande (2008) cluster sensors and discretize readings. Then they apply compression techniques on descretized data to detect correlations between sensors within each cluster. Using compression-based correlations, they fit joint distributions for variables in each cluster, and they incorporate the joint distributions in a probabilistic model, which they refer as decomposable model, to predict some sensors instead of reading them.

Deshpande et al. (2004) also apply a model based predictive approach to saving energy. They train a multivariate

Gaussian distribution which models a wireless sensor network and let user run queries on their model with arbitrary confidence intervals. Depending on target query and confidence interval, their model decides which sensors to observe. Their approach involves the entire set of sensors as one multivariate distribution and they incorporate only observations of the prediction time. In our case, we make use of current observations and all past observations.

In addition to modeling sensor networks for energy optimization, another approach is scheduling sensors for reading. Slijepcevic and Potkonjak (2001) addressed this approach to seek an efficient way of clustering sensors so that each cluster alone covers the entire area of surveillance, and that clusters alternate for sensing one at a time. They turn this problem into an area cover problem and maximize the count of clusters so that they can keep as many sensors silent as possible at each turn.

Gaussian Bayesian networks are so far used in various contexts. Castillo et al. (1997) designed a model for damage assessment in concrete structures of buildings using Gaussian Bayesian network modeling. They used their model in numerical propagation of uncertainty in incremental form and they also proved that conditional means and variances of nodes in their network are rational functions given evidence. Again Castillo, Menéndez, and Sánchez-Cambronero (2008) used Gaussian Bayesian networks to model traffic flow.

Active inference was previously applied by Bilgic and Getoor (2009; 2010) for general graphs. They used active inference to query the labels of a small number of carefully chosen nodes in a network to condition the underlying model on these collected labels to increase prediction performance on the remaining nodes. Active inference was used by Chen et al. (2011) to analyze short chunks of video on which the underlying model can condition and make better predictions on remaining segments. Bilgic and Getoor (2010) used active inference along with an iterative classification algorithm. Chen et al. (2011) used it on hidden Markov models. Active inference was also discussed in the context of hidden Markov models by Krause and Guestrin (2009). Finally Krause and Guestrin, (2005b; 2005a; 2009), and Bilgic and Getoor (2011) formulated active inference as optimal value of information on graphical models.

Limitations and Future Directions

In this work, the dGBn model of the sensor network utilized only the correlations between the sensor readings, i.e., the Y_t^s variables, ignoring the local attributes, X_t^s. Therefore, when no observation was made, the full network defaulted to the average prediction. In the future, we plan on incorporating local attribute values into dBGn so that prediction accuracy can be maximized, especially when the observed node ratio is small.

We utilized several active inference approaches in this paper, including sliding-window and impact-based approaches. There are other approaches that can be utilized as baselines. One such baseline is the variance approach where sensor readings with the highest prediction variance is chosen for observation. The motivation is that the higher the prediction variance is, the higher the chance of incorrect prediction is.

Finally, nodes in a sensor network often sense more than one measure. For example, in the Intel Research Lab data (Deshpande et al. 2004) that we used, the nodes sense temperature, humidity, light, and voltage. We focused on only the temperature readings in this paper. Training one Bayesian network model over multiple types of sensing, such as temperature and humidity, will enable the model to exploit correlations between different types of readings. For example, it is quite conceivable that humidity and temperature readings are correlated and hence it makes perfect sense to exploit these relationships.

Conclusions

We tackled the problem of simultaneously minimizing the battery consumption in a wireless sensor network by limiting how frequently the sensors can communicate their readings to a central server and minimizing prediction error over the not-communicated ones. We presented two predictive models, a Gaussian process model, and a dynamic Gaussian Bayesian network model, and several active inference approaches that selectively gather sensing information from nodes of a sensor network. We showed that by utilizing the dynamic Gaussian Bayesian network model that exploits the spatio-temporal correlations between sensor readings and performing active inference formulated through the edge weights on the Bayesian network model, we were able to reduce the prediction error drastically.

Acknowledgements This material is based upon work supported by the National Science Foundation under grant no. 1125412.

References

Ageev, A. A., and Sviridenko, M. I. 1999. Approximation algorithms for maximum coverage and max cut with given sizes of parts. In *Integer Programming and Combinatorial Optimization*. Springer. 17–30.

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

Bilgic, M., and Getoor, L. 2009. Reflect and correct: A misclassification prediction approach to active inference. *ACM Transactions on Knowledge Discovery from Data* 3(4):1–32.

Bilgic, M., and Getoor, L. 2010. Active inference for collective classification. In *Twenty-Fourth Conference on Artificial Intelligence (AAAI NECTAR Track)*, 1652–1655.

Bilgic, M., and Getoor, L. 2011. Value of information lattice: Exploiting probabilistic independence for effective feature subset acquisition. *Journal of Artificial Intelligence Research (JAIR)* 41:69–95.

Castillo, E.; Gutiérrez, J.; Hadi, A.; and Solares, C. 1997. Symbolic propagation and sensitivity analysis in gaussian bayesian networks with application to damage assessment. *Artificial Intelligence in Engineering* 11(2):173 – 181.

Castillo, E.; Menéndez, J. M.; and Sánchez-Cambronero, S. 2008. Predicting traffic flow using bayesian networks. *Transportation Research Part B: Methodological* 42(5):482 – 509.

Chen, D.; Bilgic, M.; Getoor, L.; Jacobs, D.; Mihalkova, L.; and Yeh, T. 2011. Active inference for retrieval in camera networks. In *Workshop on Person Oriented Vision*.

Cooper, G. F., and Herskovits, E. 1992. A bayesian method for the induction of probabilistic networks from data. *Machine learning* 9(4):309–347.

Deshpande, A.; Guestrin, C.; Madden, S. R.; Hellerstein, J. M.; and Hong, W. 2004. Model-driven data acquisition in sensor networks. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, 588–599. VLDB Endowment.

Elnahrawy, E., and Nath, B. 2004. Context-aware sensors. In *Wireless Sensor Networks*. Springer. 77–93.

Gilks, W. R.; Richardson, S.; and Spiegelhalter, D. J. 1996. *Markov Chain Monte Carlo in Practice*. Interdisciplinary Statistics. Chapman & Hall/CRC.

Hastings, W. K. 1970. Monte carlo sampling methods using markov chains and their applications. *Biometrika* 57(1):97–109.

Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press.

Krause, A., and Guestrin, C. 2005a. Near-optimal nonmyopic value of information in graphical models. In *Annual Conference on Uncertainty in Artificial Intelligence*.

Krause, A., and Guestrin, C. 2005b. Optimal nonmyopic value of information in graphical models - efficient algorithms and theoretical limits. In *International Joint Conference on Artificial Intelligence*, 1339–1345.

Krause, A., and Guestrin, C. 2009. Optimal value of information in graphical models. *Journal of Artificial Intelligence Research* 35:557–591.

Rasmussen, C. E. 2004. Gaussian processes in machine learning. In *Advanced lectures on machine learning*. Springer. 63–71.

Settles, B. 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool.

Slijepcevic, S., and Potkonjak, M. 2001. Power efficient organization of wireless sensor networks. In *Communications, 2001. ICC 2001. IEEE International Conference on*, volume 2, 472–476 vol.2.

Wang, L., and Deshpande, A. 2008. Predictive modeling-based data collection in wireless sensor networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4913 LNCS:34–51.