Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2011, Article ID 928958, 8 pages
doi:10.1155/2011/928958

Research Article

Comparison of Probabilistic Chain Graphical Model-Based and Gaussian Process-Based Observation Selections for Wireless Sensor Scheduling

Qi Qi and Yi Shang

Department of Computer Science, University of Missouri, Columbia, MO 65211, USA

Correspondence should be addressed to Yi Shang, shangy@missouri.edu

Received 19 March 2011; Revised 10 June 2011; Accepted 10 August 2011

The constrained power source given by batteries has become one of the biggest hurdles for wireless sensor networks to prevail. A common technique to reduce energy consumption is to put sensors to sleep between duties. It leads to a tradeoff between making a fewer number of observations for saving energy while obtaining sufficient and more valuable sensing information. In this paper, we employ two model-based approaches for tackling the sensor scheduling problem. The first approach is to apply our corrected VoIDP algorithm on a chain graphical model for selecting a subset of observations that minimizes the overall uncertainty. The second approach is to find a selection of observations based on Gaussian process model that maximizes the entropy and the mutual information criteria, respectively. We compare their performances in terms of predictive accuracies for the unobserved time points based on their selections of observations. Experimental results show that the Gaussian process model-based method achieves higher predictive accuracy if sensor data are accurate. However, when observations have errors, its performance degrades quickly. In contrast, the graphical model-based approach is more robust and error tolerant.

1. Introduction

The technology of wireless sensor networks has been popular for more than a decade in both academia and industry. Through the observations obtained by tiny embedded wireless sensors, we can have a better understanding of the natural environment, human activities, and their interactions. Researchers have been trying to turn the current relatively small-scale wireless sensor networks to a future generation of large-scale, energy-sustainable, and extensively long-standing deployments. One of the biggest hurdles is the constrained power source. Today's wireless sensors are mostly battery powered. It is not a viable way for human intervention to replace the batteries for thousands of wireless sensor nodes. The high maintenance overhead prevents the wireless sensing technology from prevailing in real-world large-scale applications.

A common technique for extending the lifetime of a wireless sensor network is to reduce its duty cycles, that is, a sensor wakes up for a small amount of time in a fixed period of interval to sense and go to sleep the rest of the time. Besides reducing sensing times for saving energy, it would be desirable to have the informative values obtained from the sensing data as high as possible. The observations should be worth of their energy costs. This is even more important for the next generation of scalable wireless networked sensors.

Tomorrow's sensors will be much smaller and energy sustainable. They can harvest energy from the environment [1]. The ambient power sources in their surrounding such as heats, mechanical movements, electromagnetic induction, and electrochemical reactions make sensing more sustainable and hassle free. However, energy harvesting on wireless sensors also brings a lot of challenges. First of all, the ambient power supply is often intermittent, and the smaller the sensor nodes are, the less power they can store. Moreover, the harvested energy usually needs to be accumulated to reach a certain level before being capable of performing some operations like taking a sensor reading or transmitting a packet. In such situations, the energy harvested sedulously should never be taken for granted. Instead, every sensing observation paid by the harvested energy should be as rewarding as possible.

Making optimal scheduling of observations with a limited power budget to maximize their information gains has become an important problem in real-world applications. It is a trade-off between obtaining more and useful information versus making less observations. The optimization of selecting observations can be considered as a subset selection problem. For example, in a task of monitoring indoor temperature, if a wireless sensor is deployed to observe for only once per hour, and we want to turn on the sensor k times for a day, then it becomes such a *subset selection problem* that k out of the total 24 time points are chosen so that the k observations will be the best selection among the other options for having the most accurate predictions of temperature readings at the unobserved time points.

In the context of statistics, the problem of subset selection or variable selection determines a subset of variables and eliminates the rest from usually a linear regression model, in order to increase predictive accuracy or to get the "big picture" with the strongest effects of predictors [2, 3]. But the selection of variables are usually for predicting only a single variable of interest. However, in the sensor scheduling case, the subset selection needs to predict the temperature distribution that covers all the unobserved time points.

Probabilistic inference in graphical models [4] provides an effective tool to deal with the quantification of uncertainty existed in the subset selection problem. The optimal subset of observations is the one that minimizes the uncertainty of the posterior conditional probability distribution of the unobserved variables. Hidden Markov models (HMMs), the chain graphical models, are often used for modelling time series data. The posterior conditional probability of unobserved variables given observations can be efficiently computed on an HMM. The VoIDP algorithm [5] was claimed to be the first optimal algorithm for efficiently making the subset selection on the chain graphical models. The name "VoIDP" was after its usage of dynamic programming approach to optimize the information value. We corrected the original VoIDP algorithm by fixing a mistake in it [6]. In this paper, we apply the corrected version of VoIDP algorithm on an HMM to solve the subset selection problem for sensor scheduling.

Gaussian process (GP) is a generalization of linear regression based on multivariate normal distributions [7]. It is often applied to spatial monitoring problems [8, 9]. Greedy methods based on heuristics, such as entropy and mutual information, can efficiently make a selection of time points. The mutual information [10] criterion, which measures entropy reduction, was shown to have better solutions than the entropy criterion in a couple of sensor placement problems [8]. In this paper, we use the GP-based approach with both the entropy and mutual information heuristics to solve the sensor scheduling problem.

In summary, the objective of the sensor scheduling problem is to select a subset of time points to turn on a sensor and keep it off at the other time to minimize prediction errors at unobserved time points. Our contribution is employing the GP model-based selection approach and our corrected version of VoIDP algorithm based on graphical models to solve the scheduling problem and comparing their performances. The GP-based approach is more data driven than the graphical model-based approach. But the latter, such as an HMM, can capture the underlying structures of the latent variables that determine the observational values in time series. Through comparison experiments, we find out that the GP-based selection approach achieves lower prediction error than the HMM-based approach given with accurate observations. However, the HMM-based selection approach performs more stably and robustly than the GP-based approach with erroneous observations.

We will briefly describe the graphical model-based, particularly HMM-based, selection approach in Section 2, and the GP-based selection approach in Section 3. The experimental results will be presented and discussed in Section 4. Following that are the conclusions.

2. Probabilistic Graphical-Model-Based Observation Selection

A sensor's time series of observations can be modeled using a probabilistic graphical model, such as an HMM. Each observation variable at a timepoint has a distribution over some hidden states such as different time periods or some events. Any observations made on the chain graphical model will contribute to the predictions of values for other unobserved variables. But the degree of the contributions depends on the selection of observations.

The quality of a selection of observations is measured based on how the observation subset changes the shape of the probability distribution of an unobserved variable. The probability distribution of any unobserved variable conditioned on a subset of observed variables can be efficiently computed using a trained HMM. A utility reward can thus be defined by the entropy on the posterior conditional probability distribution. When a subset of observation variables is selected, an expected total reward across the entire time chain is computed. Hence, the sensor scheduling problem can be formulated as a subset selection problem on a chain graphical model as follows.

Given random variables $\mathcal{X}_{\mathcal{V}} = (\mathcal{X}_1, \ldots, \mathcal{X}_n)$, a subset of the variables, $\mathcal{X}_{\mathcal{A}} = (\mathcal{X}_{i_1}, \ldots, \mathcal{X}_{i_k})$ is observed as $\mathrm{x}_{\mathcal{A}}$. $P(\mathcal{X}_{\mathcal{V}} \mid \mathcal{X}_{\mathcal{A}} = \mathrm{x}_{\mathcal{A}})$ is the conditional observation distribution over all variables given the observation $\mathrm{x}_{\mathcal{A}}$. A total reward $R(P(\mathcal{X}_{\mathcal{V}} \mid \mathcal{X}_{\mathcal{A}} = \mathrm{x}_{\mathcal{A}}))$ is also given. The subset selection problem is

$$\mathcal{A}^* = \arg\max_{\mathcal{A}} \sum_{\mathrm{x}_{\mathcal{A}}} P(\mathcal{X}_{\mathcal{A}} = \mathrm{x}_{\mathcal{A}}) R(P(\mathcal{X}_{\mathcal{V}} \mid \mathcal{X}_{\mathcal{A}} = \mathrm{x}_{\mathcal{A}})), \quad (1)$$

where the expected total reward is used because the future observations of $\mathcal{X}_{\mathcal{A}}$ are unknown. The selections are made based on the model before knowing any observational values. Because of conditional independency on chain graphical models, the expected total reward to maximize in (1) is also the summation of all the expected local rewards, where an expected local reward $R_j(\mathcal{X}_j \mid \mathcal{X}_{\mathcal{A}})$ equals to $\sum_{\mathrm{x}_{\mathcal{A}}} P(\mathcal{X}_{\mathcal{A}} = \mathrm{x}_{\mathcal{A}}) R_j(\mathcal{X}_j \mid \mathrm{x}_{\mathcal{A}})$. A local reward $R_j(\mathcal{X}_j \mid \mathrm{x}_{\mathcal{A}})$ depends on $P(\mathcal{X}_j \mid \mathcal{X}_{\mathcal{A}} = \mathrm{x}_{\mathcal{A}})$, which is the marginal distribution of variable $\mathcal{X}_j$ conditioned on observations $\mathcal{X}_{\mathcal{A}} = \mathrm{x}_{\mathcal{A}}$. We can define the reward by using conditional entropy, that is, $R_j(\mathcal{X}_j \mid \mathrm{x}_{\mathcal{A}}) = -H(\mathcal{X}_j \mid \mathrm{x}_{\mathcal{A}}) = \int P(x_j, \mathrm{x}_{\mathcal{A}}) \log_2 P(x_j \mid \mathrm{x}_{\mathcal{A}}) \mathrm{d}x_j$.

The conditional independence property of chain graphical models simplifies the evaluation of $P(\mathcal{X}_j \mid \mathcal{X}_\mathcal{A})$. If only the observations made before a time point are used, then the evaluation of $P(\mathcal{X}_j \mid \mathcal{X}_\mathcal{A})$ only depends on $P(\mathcal{X}_j \mid \mathcal{X}_i)$, where $\mathcal{X}_i$ is the closest observation made before j. Moreover, the conditional independence property also decomposes the expected total reward for the entire chain into the local rewards for subchains that are separated by the observations. This important property inspired an efficient approximate algorithm based on the divide-and-conquer strategy [5].

The subset selection problem in (1) is a combinatorial optimization problem like the *Knapsack* problem, although its utility function is more computationally complicated. It is an NP-hard problem. The *Knapsack* problem admits a pseudopolynomial time algorithm, and all known such algorithms for NP-hard problems are based on dynamic programming [11].

The authors in [5, 12] developed an algorithm called VoIDP to solve the subset selection problem, and it was claimed to be the first optimal algorithm for selecting the optimal observations on chain graphical models. It implements a dynamic programming approach that exploits the chain structured models to efficiently evaluate the expected total reward. The time complexity of the algorithm was proved to be $(1/6n^3 + \mathcal{O}(n^2))B$ given budget B in terms of the number of evaluations of the rewards, where $n = |\mathcal{V}|$. Although it works efficiently and optimally with a chain graphical model, it is still much harder to be applied on more general forms of graphical models. It was proved that the problem of selecting a subset of observations for even discrete poly-tree is computational intractable.

We found out a mistake in the originally published VoIDP algorithm that fails to give correct solution. We corrected the mistake and presented an improved version of the algorithm in [6]. In the experiments, we will use the corrected VoIDP algorithm that will be briefly presented in what follows. Once the optimal observations are selected, the predictive value will be calculated by taking the expected mean of a posterior conditional observation distribution.

Algorithm 1 shows the corrected version of VoIDP. The first part of the algorithm dynamically computes a three-dimension $(a, b,$ and $k)$ table, where a and b are the two ends of a chain, and k is the budget. Each cell in the table is a tuple of two elements: $L_{a:b}(k)$ and $\Phi_{a:b}(k)$. $L_{a:b}(k)$ represents the optimal expected total reward of the subchain $a : b$ with the budget k. $sel(j)$ is the expected total reward of the subchain $a : b$ obtained by observing at time point j, and $sel(0)$ is the reward when no observation is made. $\Phi_{a:b}(k)$ stores the j that maximizes the value of $sel(j)$. $L_{a:b}(0)$ is the base case, and the recursion for $L_{a:b}(k)$ is either $L_{a:b}(0)$ or $\max_{a<j<b,\, \beta_j \le k}\{sel(j)\}$. It means that we can choose not to spend any more of the budget to reach the base case, or select the optimal observation at j with the cost β_j. C_j denotes penalty on the reward by selecting at j. In our experiments, we let C_j be zero and β_j be one.

The second part of Algorithm 1 computes the optimal selections by tracking through the values of $\Phi_{a:b}(k)$. Initially a and b are set to represent the entire chain. It starts with the full budget B and the empty selection set $\mathcal{A}$. In the following loop, it finds the optimal selection y from $\Phi_{a:b}(k)$, and adds it into the selection set $\mathcal{A}$. Whenever a selection is made, it spent a budget of β_y and cuts the chain into two parts, $a : y$ and $y : b$. The searching of optimal observation will be continued on the second subchain $y : b$. This process will stop when the budget is used up. The bold-faced $a \leftarrow y$ is important because it makes the tracking to fall into the correct domain in the whole table, but it was not in the original VoIDP algorithm.

```
Input: Budget $B$, rewards $R_j$, costs $\beta_j$ and penalties $C_j$
      $(j \in \mathcal{V}, |\mathcal{V}| = n)$
Output: Optimal selection of observations at $\mathcal{A}$
begin
  for $0 \le a < b \le n+1$ do compute $L_{a:b}(0)$
  for $k = 1$ to $B$ do
      for $0 \le a < b \le n+1$ do
          $sel(0) \leftarrow L_{a:b}(0)$
          for $j = a+1$ to $b-1$ do
              $sel(j) \leftarrow$
              $R_j(\mathcal{X}_j \mid \mathcal{X}_j) - C_j + L_{a:j}(0) + L_{j:b}(k - \beta_j)$
          $L_{a:b}(k) \leftarrow \max_{j \in \{0,\, a+1,\dots,b-1\}} sel(j)$
          $\Phi_{a:b}(k) \leftarrow \arg\max_{j \in \{0,a+1,\dots,b-1\}} sel(j)$
  $a \leftarrow 0; b \leftarrow n+1$
  $k \leftarrow B$
  $\mathcal{A} \leftarrow \emptyset$
  repeat
    $y \leftarrow \Phi_{a:b}(k)$
    if $y > 0$ then
        $\mathcal{A} \leftarrow \mathcal{A} \cup \{y\}$
      a ← y
      $k \leftarrow k - \beta_y$
    else
      break
  until $k \le 0$
```

Algorithm 1: Corrected VoIDP algorithm for optimizing observation selection on chain graphical models.

3. Gaussian Process-Based Observation Selection

A Gaussian process (GP), by definition [7], is a collection of random variables, and any finite number of variables in the collection also has a joint Gaussian distribution. For scheduling a sensor, we assume that the joint probability of its observations at all the time points is a multivariate Gaussian distribution:

$$P(\mathcal{X}_\mathcal{V} = \mathrm{x}_\mathcal{V}) = \frac{1}{(2\pi)^{n/2}|\Sigma_{\mathcal{V}\mathcal{V}}|^{1/2}} e^{-1/2(\mathrm{x}_\mathcal{V}-\mu_\mathcal{V})^T \Sigma_{\mathcal{V}\mathcal{V}}^{-1}(\mathrm{x}_\mathcal{V}-\mu_\mathcal{V})}, \tag{2}$$

where $\mathcal{V}$ denotes the whole set of the variable indices with $|\mathcal{V}| = n$. $\mu_\mathcal{V}$ is the mean vector and $\Sigma_{\mathcal{V}\mathcal{V}}$ is the covariance matrix. A subset $\mathcal{A}$ from $\mathcal{V}$ also satisfies that $\mathcal{X}_\mathcal{A} \sim \mathcal{N}(\mu_\mathcal{A}, \Sigma_{\mathcal{A}\mathcal{A}})$, where $\mu_\mathcal{A}$ is a subvector of $\mu_\mathcal{V}$ and $\Sigma_{\mathcal{A}\mathcal{A}}$ is the relevant submatrix of $\Sigma_{\mathcal{V}\mathcal{V}}$. This consistency property is also known as the marginalization property.

The nice property also holds in $P(\mathcal{X}_i \mid \mathcal{X}_{\mathcal{A}} = \mathrm{x}_{\mathcal{A}})$, the probability distribution of $\mathcal{X}_i$ conditioned on the observational values $\mathrm{x}_{\mathcal{A}}$ of a selected variable set $\mathcal{X}_{\mathcal{A}}$. This posterior conditional probability is also a Gaussian distribution with conditional mean $\mu_{i|\mathcal{A}}$ and variance $\sigma^2_{i|\mathcal{A}}$ given by

$$\mu_{i|\mathcal{A}} = \mu_i + \Sigma_{i\mathcal{A}}\Sigma^{-1}_{\mathcal{A}\mathcal{A}}(\mathrm{x}_{\mathcal{A}} - \mu_{\mathcal{A}}), \tag{3}$$

$$\sigma^2_{i|\mathcal{A}} = \Sigma_{ii} - \Sigma_{i\mathcal{A}}\Sigma^{-1}_{\mathcal{A}\mathcal{A}}\Sigma_{\mathcal{A}i}, \tag{4}$$

where $\mu_{\mathcal{A}}$ is the mean vector of variable set $\mathcal{X}_{\mathcal{A}}$, $\Sigma_{i\mathcal{A}}$, and $\Sigma_{\mathcal{A}i}$ are the corresponding submatrices of $\Sigma_{\mathcal{V}\mathcal{V}}$. For instance, the $\Sigma_{i\mathcal{A}}$ is calculated by taking the ith row and the $\mathcal{A}$ columns of $\Sigma_{\mathcal{V}\mathcal{V}}$. μ_i and Σ_{ii} can be considered as the prior mean and the prior variance of the observations at i.

3.1. The Entropy Heuristic. The conditional probability $P(\mathcal{X}_i \mid \mathcal{X}_{\mathcal{A}})$ carries very important information for evaluating the quality of the selection set $\mathcal{A}$, which can be measured by the conditional entropy:

$$\begin{aligned} H(\mathcal{X}_i \mid \mathcal{X}_{\mathcal{A}}) &= -\int\int P(x_i, x_{\mathcal{A}}) \log P(x_i \mid x_{\mathcal{A}}) \mathrm{d}x_i \mathrm{d}x_{\mathcal{A}} \\ &= \frac{1}{2}\log \sigma^2_{i|\mathcal{A}} + \frac{1}{2}(\log \pi + \log 2 + 1), \end{aligned} \tag{5}$$

where $P(\mathcal{X}_i \mid \mathcal{X}_{\mathcal{A}})$ is assumed as a Gaussian probability distribution. Note that the entropy is a monotonic function of its variance $\sigma^2_{i|\mathcal{A}}$, which can be evaluated ahead of making any observations (4).

The sensor scheduling problem becomes to select a subset of time points at $\mathcal{A}$ (out of the variable index set $\mathcal{V}$) to turn on the sensor and to keep it off at all the other time (as of indices in $\mathcal{V} \setminus \mathcal{A}$). The subset selection can be optimized by minimizing the entropy $H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_{\mathcal{A}})$. This is also equivalent to finding a subset $\mathcal{A}$ that maximizes $H(\mathcal{X}_{\mathcal{A}})$, as the chain rule for conditional entropy holds that $H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_{\mathcal{A}}) = H(\mathcal{X}_{\mathcal{V}}) - H(\mathcal{X}_{\mathcal{A}})$. The optimization problem turns out to be an NP-hard problem. To solve it, a greedy heuristic is to iteratively find the next selection $y^*_{i+1} \in \mathcal{V} \setminus \mathcal{A}_i$ that has the highest conditional entropy given the current selection set $\mathcal{A}_i$:

$$y^*_{i+1} = \arg\max_{y_{i+1}} H\Big(\mathcal{X}_{y_{i+1}} \mid \mathcal{X}_{\mathcal{A}_i}\Big). \tag{6}$$

Algorithm 2 shows the greedy algorithm based on the entropy heuristic. k is the size of the selection and the $\sigma^2_{y|\mathcal{A}}$ can be computed using (4). Because the log function is monotonic, $\sigma^2_{y|\mathcal{A}}$ is proportional to $H(\mathcal{X}_y \mid \mathcal{X}_{\mathcal{A}})$. That means finding a selection at y that maximizes $H(\mathcal{X}_y \mid \mathcal{X}_{\mathcal{A}})$ is equivalent to finding such a y that maximizes $\sigma^2_{y|\mathcal{A}}$.

The computation of $\sigma^2_{y|\mathcal{A}}$ is expensive. Let $|\mathcal{V}| = n$, there are n times of these computations when $i = 1$, and $(n-k+1)$ times when $i = k$. Hence, Algorithm 2 has totally $(2n - k + 1)k/2$ times of evaluations of $\sigma^2_{y|\mathcal{A}}$.

```
Input: covariance matrix $\Sigma_{\mathcal{V}\mathcal{V}}$, selection size $k$
Output: selection set $\mathcal{A}(\mathcal{A} \subseteq \mathcal{V}$, and$|\mathcal{A}| = k)$
begin
  $\mathcal{A} \leftarrow \emptyset$
  for $i = 1$ to $k$ do
    foreach $y \in \mathcal{V} \setminus \mathcal{A}$ do $\delta_y \leftarrow \sigma^2_{y|\mathcal{A}}$
    $y^* \leftarrow \arg\max_{y \in \mathcal{V}\setminus\mathcal{A}} \delta_y$
    $\mathcal{A} \leftarrow \mathcal{A} \cup \{y^*\}$
```

ALGORITHM 2: Greedy algorithm for maximizing entropy $H(\mathcal{A})$.

3.2. The Mutual Information Heuristic. Another criterion for optimizing the subset selection is the mutual information, which was originally proposed by Caselton and Zidek in [10]. The mutual information of a subset $\mathcal{A}$ denoted as MI($\mathcal{A}$) is defined as follows, and it is actually the entropy reduction:

$$\begin{aligned} MI(\mathcal{A}) &= I(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}; \mathcal{X}_{\mathcal{A}}) \\ &= H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}) - H(\mathcal{X}_{\mathcal{V}\setminus\mathcal{A}} \mid \mathcal{X}_{\mathcal{A}}) \\ &= H(\mathcal{X}_{\mathcal{A}}) - H(\mathcal{X}_{\mathcal{A}} \mid \mathcal{X}_{\mathcal{V}\setminus\mathcal{A}}). \end{aligned} \tag{7}$$

The authors in [8] demonstrated the advantage of the mutual information criterion over the entropy criterion in sensor placement for a couple of spatial monitoring applications. They found out that the mutual information criterion led to a more intuitive sensor placement than the entropy criterion did. The mutual information criterion placed sensors in the central areas of a sensing space whereas the entropy placed sensors mostly at the boundaries.

In contrast to the entropy-based selection maximizing only the uncertainty of the selection set $\mathcal{A}$, the mutual information criterion maximizes the reduction of the entropy over the rest of the variable space $\mathcal{V} \setminus \mathcal{A}$ before and after observing at $\mathcal{A}$. That is, to schedule a sensor, we will find a subset of time points at $\mathcal{A}^*$ such that

$$\mathcal{A}^* = \arg\max_{\mathcal{A}\subset\mathcal{V}} MI(\mathcal{A}). \tag{8}$$

Optimization of the mutual information is also an NP-complete problem. A greedy algorithm was developed in [8] that selects the next variable y maximizing the mutual information gain:

$$\Delta_y = MI(\mathcal{A} \cup y) - MI(\mathcal{A}). \tag{9}$$

The greedy heuristic chooses the next selection that provides the maximal increase in the values of mutual information.

In the context of Gaussian process and based on the entropy equation in (5), (9) can be further deduced as:

$$\begin{aligned} \Delta_y &= MI(\mathcal{A} \cup y) - MI(\mathcal{A}) \\ &= H(y \mid \mathcal{A}) - H\Big(y \mid \overline{\mathcal{A}}\Big) \\ &= \frac{1}{2}\log_2\left(\frac{\sigma^2_{y|\mathcal{A}}}{\sigma^2_{y|\overline{\mathcal{A}}}}\right), \end{aligned} \tag{10}$$

where $\overline{\mathcal{A}}$ means all the variable indices in $\mathcal{V}$ excluding $\mathcal{A}$ and y, which can also be denoted as $\mathcal{V} \setminus (\mathcal{A} \cup y)$.

```
Input: covariance matrix $\Sigma_{\mathcal{V}\mathcal{V}}$, selection size $k$
Output: selection set $\mathcal{A}(\mathcal{A} \subseteq \mathcal{V})$, mutual information
        gains $\Delta$
begin
  $\mathcal{A} \leftarrow \emptyset$
  foreach $y \in \mathcal{V}$ do $\Delta_y \leftarrow +\infty$; $\Phi_y \leftarrow 0$
  for $i = 1$ to $k$ do
      repeat
        $y^* \leftarrow \arg\max_{y \in \mathcal{V} \setminus \mathcal{A}} \Delta_y$
        if $\Phi_{y^*} == i$ then
          break
        else
            $\overline{\mathcal{A}} \leftarrow \mathcal{V} - (\mathcal{A} \cup y^*)$
            $\Delta_{y^*} \leftarrow 1/2 \log_2(\sigma^2_{y^*|\mathcal{A}} / \sigma^2_{y^*|\overline{\mathcal{A}}})$
            $\Phi_{y^*} \leftarrow i$
      until 0
      $\mathcal{A} \leftarrow \mathcal{A} \cup \{y^*\}$
```

Algorithm 3: Greedy algorithm for maximizing mutual information gain $MI(\mathcal{A} \cup y) - MI(\mathcal{A})$ using lazy evaluation.

An interesting note about the mutual information gain Δ_y is that it is monotonically decreasing as the selection set $\mathcal{A}$ gets larger. It inspired the enhanced greedy algorithm with lazy evaluation [8].

The greedy algorithm with lazy evaluation for maximizing the mutual information is presented in Algorithm 3. We rewrite it as in the context of the Gaussian process model.

Φ_{y^*} records in which iteration Δ_{y^*} is updated. The lazy evaluation saves a lot of computation of Δ_y based on the insight that the sequence of the mutual information gains on a fixed y decreases as set $\mathcal{A}$ gets bigger. It will select the y^* if the maximal Δ_{y^*} is updated in the current iteration, otherwise it will update Δ_{y^*} and Φ_{y^*} and repeat the selection process.

When $|\mathcal{V}| = n$, Algorithm 3 has $2(n + k - 1)$ times of evaluations of either $\sigma^2_{y^*|\mathcal{A}}$ or $\sigma^2_{y^*|\overline{\mathcal{A}}}$ in the best case. This is more efficient and scalable than Algorithm 2 when n becomes very large.

Algorithm 3 is not only efficient, but also provides its solution with a theoretic bound to the optimal solution. Although the mutual information function in (7) is not always increasing, the authors in [8] proved that it is still a partially monotonic submodular function. According to [13], a greedy algorithm, such as the Algorithm 3, which optimizes a monotonic submodular function guarantees a theoretical performance lower bound of $(1 - 1/e)$OPT, where OPT is the optimal solution value.

After a subset of observations at $\mathcal{A}$ is selected, the conditional probability distribution of an unobserved variable i given the observations of $\mathcal{X}_{\mathcal{A}}$ can be computed as $P(\mathcal{X}_i \mid \mathcal{X}_{\mathcal{A}})$. We will use its mean computed by (3) as the predictive value of the observation variable $\mathcal{X}_i$.

4. Comparison Experiments

In this section, we will compare the probabilistic graphical model-based approach and the Gaussian process model-based approach in solving the subset selection problem for scheduling a sensor. Specifically, we want to select a subset of time points in size k out of totally 24 time points for turning on a sensor to sense and keeping it off at the other time. For convenience, in the following experiments, we will refer to the probabilistic graphical model-based approach simply as the HMM-based method, and the Gaussian Process-based approach simply as the GP-based method. The performance is measured using predictive accuracy for the unobserved time points in terms of the root mean squares (RMS) error. The methods are compared for accurate observations, as well as erroneous observations.

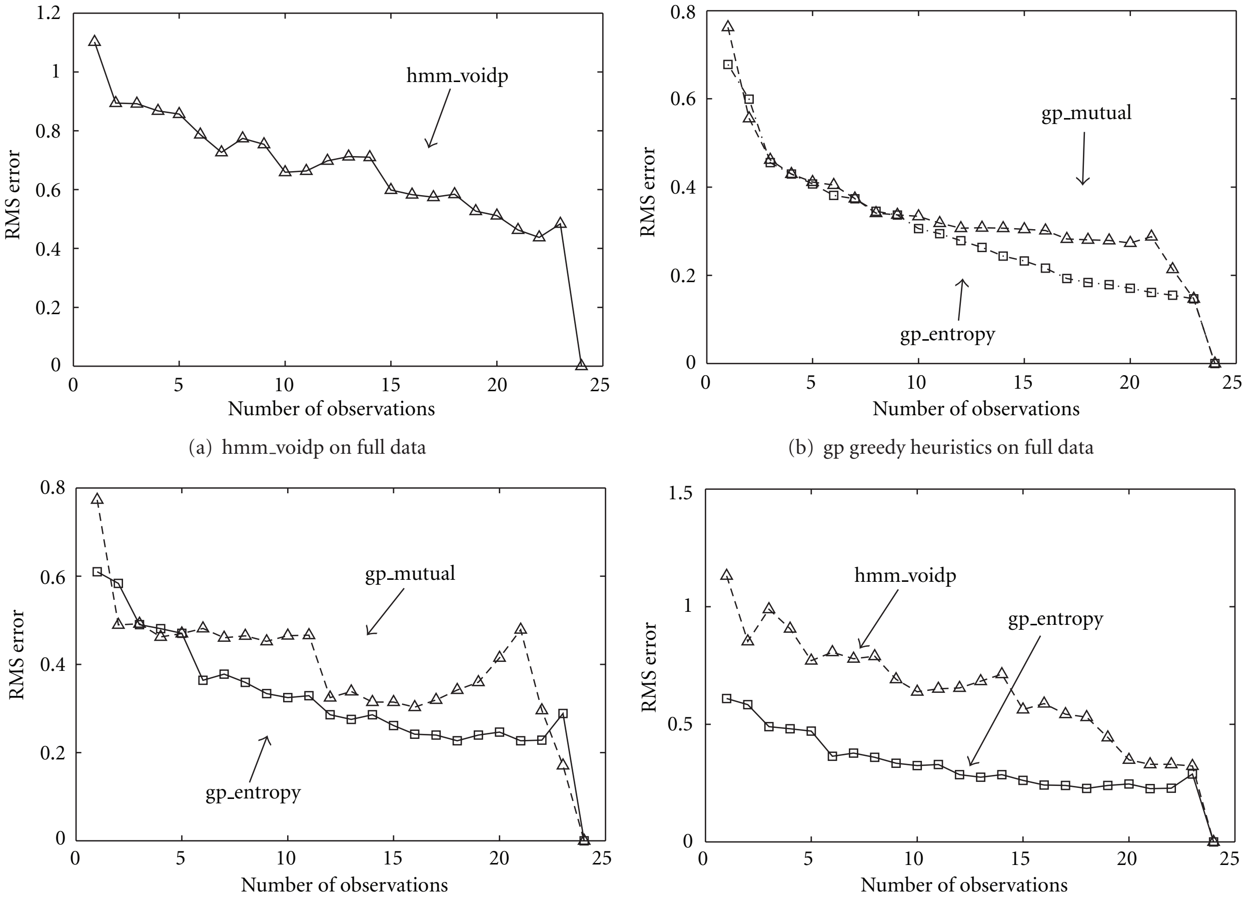
4.1. Experimental Setup. A hidden Markov model and a multivariate Gaussian model were trained using the temperature time series data collected in the Intel Berkeley Research Laboratory [14]. All the data were preprocessed for missing samples and discretized into 10 bins of 2 degrees Kelvin. The full dataset consists of temperature samples combined from three neighbored sensors for 19 days. When training the chain graphical model, we set four latent states representing the different time periods 12 am–7am, 7 am–12 pm, 12 pm–7 pm, and 7 pm–12 am. The whole dataset was also randomly split into the test set and the training set with the ratio 1 : 9. A small error-injected test dataset was also generated. The errors were taken randomly from a normal distribution with mean zero and variance 0.25.

We use these notations in the figures: "hmm_voidp" represents the HMM-based selection approach by our improved VoIDP algorithm; "gp_entropy" represents the GP-based selection approach by employing the entropy heuristic, and "gp_mutual" for the GP-based mutual information heuristic.

4.2. Results and Discussion. Figures 1(a) and 1(b) show the results on the full dataset. Generally speaking, the more observations are selected the less RMS prediction errors are achieved for both the HMM based and the GP based approaches. In Figure 1(b), the mutual information heuristic holds the competition with the entropy heuristic until about 10 observations are selected. We have mentioned that the mutual information function is not always monotonically increasing as the selection set gets bigger. The mutual information gains and its values are shown in Figures 2 and 3, respectively. It can be seen that the mutual information heuristic lost its advantage after 12 observations are selected. It explains why the mutual information heuristic only has an advantage over the entropy heuristic between the 2 and 5 observations, but loses afterwards on the test dataset, as shown in Figure 1(c).

The GP entropy-based selection is compared with the HMM-based selection on the test data in Figure 1(d). The GP-based approach beats the HMM-based approach in terms of predictive accuracy on the test data.

Figures 4(a)–4(c) examine how the erroneous observations affect these model-based selection approaches. In the figures, the circles and crosses denote the results, the dash and solid lines present the trends by doing regression based on the results. It shows that both GP based heuristic methods



(a) hmm_voidp on full data

(b) gp greedy heuristics on full data

(c) gp greedy heuristics on test data

(d) hmm_voidp versus gp_entropy on test data

Figure 1: Prediction error versus number of selected observations on full dataset (a, b), and on test dataset (c, d), given by the HMM-based and GP-based selection approaches.

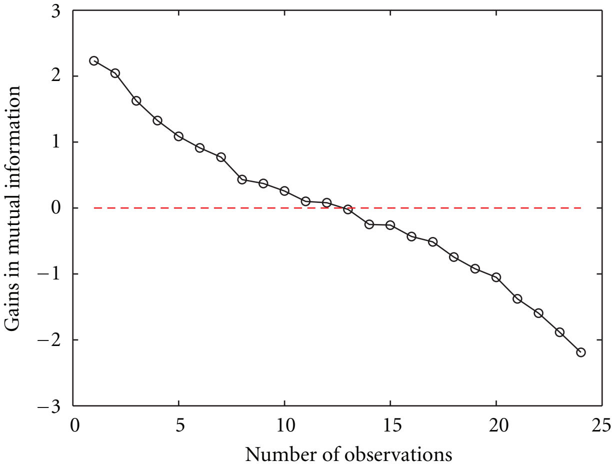


Figure 2: Mutual information gains on observations.

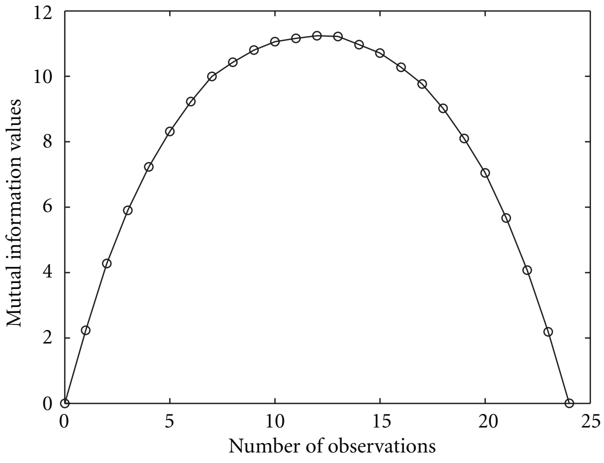
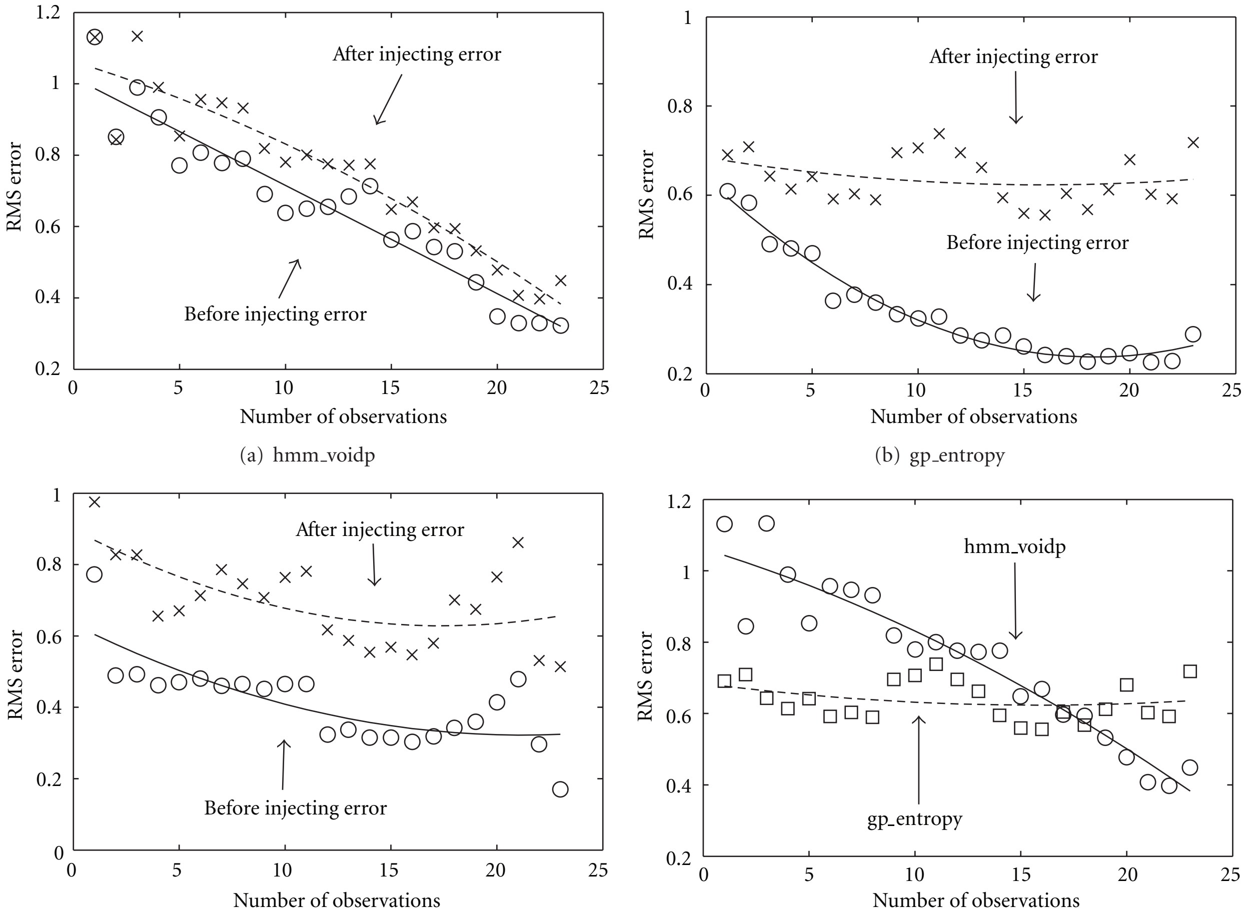


Figure 3: Mutual information values on observations.

suffer big losses in predictive accuracy given the error-injected observations. However, the HMM-based approach shows a very robust performance with a little increased of the prediction errors.



(a) hmm_voidp

(b) gp_entropy

(c) gp_mutual

(d) hmm_voidp versus gp_entropy on error injected test data

Figure 4: (a–c) Comparison of prediction errors on the original and error-injected test data by HMM-based selection, GP-based entropy heuristic, and GP-based mutual information heuristic selections, respectively; (d) comparing HMM-based selection against GP-based entropy heuristic selection on the test data with erroneous observations.

Despite both the GP-based heuristics perform poorly with erroneous observations, the mutual information heuristic is slightly more stable. In Figure 4(c), the two trend lines are parallel to each other whereas Figure 4(b) shows a big difference in performances by the entropy heuristic before and after the errors are injected into the observations.

The GP-based entropy heuristic is compared with the HMM-based selection under the erroneous observations because of its relatively lower prediction error over the mutual information heuristic. Figure 4(d) illustrates the result. It shows that the GP-based entropy method tends to maintain a constant error as the number of observations increases, meaning more observations do not help the predictive accuracy. Conversely, the HMM-based approach has a decreasing trend on predictive accuracy as more observations are added. After 16 observations the HMM-based selection method outperforms the GP-based method in terms of the RMS prediction errors.

The robustness against observation errors exhibited by the HMM-based selection approach is partly attributed to the conditional independence property provided inherently by the corresponding probabilistic chain graphical model. The observations on the chain graphical model cut the entire chain into smaller subchains, and the observations on one subchain will not directly affect the predictions on the other subchains. This property minimizes the effect caused by the erroneous observations. But for the GP-based approach, the predictions take all the observations directly into account, which therefore increases the chance of being affected by the errors in observations.

5. Conclusions

In this paper, we tackle the sensor scheduling problem by selecting a subset of time points to observe in order to make the most accurate predictions for the unobserved time points. We compare two model-based selection approaches, the probabilistic graphical model-based approach, particularly with HMM, and the Gaussian process model-based approach employing both entropy and mutual information greedy heuristics.

The results show that the GP-based approach performs better than the graphical model-based approach in terms of the predictive accuracy with accurate observations. But when small errors are injected into the observations, the GP-based selection method performs very poorly. In contrast, the graphical model-based approach demonstrates more robust and stable performance given the erroneous observations, and outperforms the GP-based approach when more observations are selected.

References

[1] P. Dutta, "Sustainable sensing for a smarter planet," *XRDS*, vol. 17, pp. 14–20, 2011.

[2] E. I. George, "The Variable Selection Problem," *Journal of the American Statistical Association*, vol. 95, no. 452, pp. 1304–1308, 2000.

[3] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, NY, USA, 2nd edition, 2009.

[4] M. I. Jordan, *Learning in Graphical Models*, MIT Press, Cambridge, Mass, USA, 2nd edition, 2001.

[5] A. Krause and C. Guestrin, "Optimal value of information in graphical models," *Journal of Artificial Intelligence Research*, vol. 35, no. 1, pp. 557–591, 2009.

[6] Q. Qi, Y. Shang, and H. Shi, "An improved algorithm for optimal subset selection in chain graphical models," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '10)*, 2010.

[7] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, Mass, USA, 2006.

[8] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, pp. 235–284, 2008.

[9] P. Zhuang, Q. Qi, Y. Shang, and H. Shi, "Model-based traffic prediction using sensor networks," in *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC '08)*, pp. 136–140, 2008.

[10] W. F. Caselton and J. V. Zidek, "Optimal monitoring network designs," *Statistics and Probability Letters*, vol. 2, no. 4, pp. 223–227, 1984.

[11] V. V. Vazirani, *Approximation Algorithms*, Springer, Berlin, Germany, 2nd edition, 2003.

[12] A. Krause and C. Guestrin, "Optimal nonmyopic value of information in graphical models: efficient algorithms and theoretical limits," in *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI '05)*, pp. 1339–1345, Morgan Kaufmann Publishers, 2005.

[13] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[14] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB '04)*, pp. 588–599, VLDB Endowment, 2004.