Sensor Reading Prediction using Anisotropic Kernel Gaussian Process Regression

Erliyah Nurul Jannah
Dept. of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
Email: erliyah.nj@gmail.com

Hsing-Kuo Pao
Dept. of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan
Email: pao@mail.ntust.edu.tw

***Abstract*—We utilize sensors to help us monitor events in the environment around us. To save power consumption, we often prefer to use as few sensors as possible and the sensors can be on for as limited time as possible while keeping the same or similar service performance from the sensors. In this work, we propose a mechanism that can use a small subset of sensor readings and the rest of sensor readings that are not collected can be approximated by the available sensor readings. We adopt Gaussian process regression as the prediction model. One key to have an effective Gaussian process prediction given sensor reading data of high variety relies on how we find an appropriate kernel function for the process. More specifically, given sensor data that have spatial and temporal relationships, we propose an anisotropic kernel for the process that can integrate different relationships as one and we can successfully describe the relationship between each pair of different sensor readings for the reading prediction. The experiments for evaluation are conducted based on a case study on weather data that consist of temperature readings collected in Taiwan. The experiment results show that the proposed Gaussian process regression with anisotropic kernel function can well describe the spatio-temporal relationships between different sensor readings and give effective temperature prediction.**

***Keywords*-Anisotropic kernel, Gaussian process regression, spatio-temporal modeling, temperature prediction.**

I. Introduction

Sensors deployed in the environment around us can help us collect useful information for our daily lives. In Intelligent Transportation Systems (ITS), smart homes, or agriculture applications, homogeneous or heterogeneous sensors are utilized to record quantities such as temperature, humidity, GPS logs, motion acceleration, etc and we can dynamically make decisions to improve our life quality. One key for successful sensor deployment is to configure sensors so that the energy consumption is minimized [5]. To save energy, we can choose some sensors and put them to sleep when they have similar behavior to other neighboring sensors. In this work, we study the spatial and temporal relationships between different sensor readings. Based on the study, one can select a small subset of sensor readings as the representative set and the information of all sensors can be well approximated by this representative set. We adopt Gaussian process regression with an anisotropic kernel function that balances between spatial and temporal relationships for sensor reading modeling.

There are numerous energy saving approaches that have been proposed. To monitor activities in an environment, technicians deploy sensors efficiently when a small subset of sensors can well represent all the sensors in the environment. For instance, one can avoid identical sensor data obtained from different sensors. In precise, researchers aim at improving area spectral efficiency (ASE) [1] for efficient sensing. Another approach is to find a period and put the sensors to sleep [4]. Relatively stable readings can be ignored as long as we find no surprise from the readings. In summary, fewer sensors to install and less time to keep sensors on can save power consumption.

One approach to integrate the two savings: the saving of sensor usage and the saving of sensor's waking up time relies on how we can relate two sensor readings, the readings on two sensor locations or the readings collected in two moments for a single sensor. Given the relationship between each pair of sensor readings, we can apply any regression models to interpolate or extrapolate missing readings (the readings from the sensors that are turned off) given the representative readings (the readings from the sensors that stay awake) in an environment. In this work, we adopt Gaussian process regression (GPR) [6] for sensor reading interpolation and prediction. Other than predicting a single answer, GPR also provides the prediction confidence level for users to make precise decisions.

Technically, to define a Gaussian process (GP) we need to find a suitable kernel function so that the relationship between any pair of sensor readings can be well modeled. To discover the relation between sensor readings, one has to well describe the characteristics of sensor readings, such as *where*, *when* and *how* we collect the readings. In GP, thanks to the nice properties of Gaussian distributions [6], we only need to define appropriate kernel function given a pair of sensor readings[1]. A simple choice of kernel function is decided by the Euclidean distance between two readings. In a weather prediction application given temperature sensors, we can compute the Euclidean distance between two

[1]We often assume the mean function which is also necessary when we specify a Gaussian process to be a zero function.

temperature readings. We should at least consider where the temperature sensors are deployed and when we collect the temperature values. Intuitively, we expect similar readings when the readings are collected in nearby sensors or in close moments. However, the Euclidean distance may not be suitable to be used to define the relation or the kernel function between two readings in the following cases:

1) The geographic distance between two sensors may not fully reflect how sensors are related. For instance, two temperature sensor readings, even they are collected in nearby sensors may not share similar patterns if the two sensors are allocated in locations with very different elevations.
2) The sensor readings may contain various of characteristics and one could be more important than another when we need to define the relation between two readings. Technically, we say that the heterogenous characteristics need to be transformed to a normalized version so that we can compute the distance based on different characteristics.
3) The sensor readings may contain some characteristics that have a nature hierarchical structure. For instance, we may describe *time* as a $(\text{year}, \text{month}, \text{date}, \text{time})$ quadruple. To define the distance between two sensor readings with hierarchical structure, we have to choose a distance or kernel function that can take the hierarchical structure into account.

In this work, we propose an anisotropic kernel function that can deal with the sensor data of the above properties. To support our idea, we conduct a case study on a network that consists of many temperature sensors that are deployed in Taiwan. Based on our study, we can well describe the relation between two sensor readings and predict a sleeping sensor's reading given its neighboring readings. With such technique, we can choose a small sensor subset to be the awake set; also, choose a time to wake up sensors when necessary. Overall, the power consumption is saved.

The rest of the paper is organized as follows: In Section II we give a brief introduction of Gaussian process, Gaussian process regression and the kernel function that is used for the Gaussian process. After that, Section III is the case study on a set of sensors that are deployed in Taiwan for temperature monitoring. We shall discuss how the proposed anisotropic Gaussian process regression can help us for temperature prediction. The evaluation of the proposed method is shown in Section IV. The last part concludes this work.

II. Gaussian Process

Gaussian Process (GP) is a well known method to model sequential data with discrete or continuous time [6]. We can use GP to solve both of regression and classification problems. In this work, we mainly focus on GP Regression (GPR) related topics.

A. Gaussian Process Regression

Regression is a machine learning problem that aims to learn the relation between the attribute set $\mathbf{x}$ and the response $y \in \mathbb{R}$, given the input data $\mathcal{D} \equiv \{(\mathbf{x}_i, y_i) : i = 1, 2, \ldots, n\}$. Various methods have been proposed to solve the regression problems, which include linear and nonlinear types of models. GPR is a method that can give both the prediction and the confidence level that is associated with the prediction result at the same time; also, GPR can take linear or nonlinear functions as the base function in its modeling. In general, there are two kinds of machine learning approaches, the parametric and the nonparametric ones. GPR is a nonparametric type of method. Most learning tasks must be able to avoid the problems of overfitting or underfitting. GPR provides a natural mechanism so that we can flexibly choose between a simple model or a complicated model for various of inputs and applications [9].

Formally, given a data set $\mathcal{D}$ where $\mathbf{x}_i \in \mathbb{R}^d$ describes the attributes, and y_i is the scalar output of the i-th data. GPR will find the function f that can associate $\mathbf{x}_i$ to y_i by setting appropriate GP parameters. Gaussian process has good properties where we only need to specify the mean function $m(\mathbf{x})$ and the covariance function $cov(\mathbf{x}, \mathbf{x}')$ to completely decide the process. We can write:

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), cov(\mathbf{x}, \mathbf{x}')) \,. \tag{1}$$

The covariance function is one of the most important part to decide the whole GP. Very often, we have the covariance function decided by the distance between two attribute sets $\mathbf{x}$ and $\mathbf{x}'$, written as:

$$cov(\mathbf{x}, \mathbf{x}') = k(\|\mathbf{x} - \mathbf{x}'\|) \,, \tag{2}$$

for a decreasing function k. These GP properties are needed to be defined before we use GPR (prior properties). Prior properties of GP can be determined from sampled data [6], [7]. By giving this prior information, it means that we give a restriction to the candidate functions f for our final prediction. There are many kinds of covariance functions, according to different kinds of distance metrics. One of the most frequently used covariance function is the squared exponential function shown as follows,

$$cov(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}\|\mathbf{x} - \mathbf{x}'\|^2\right) + \sigma_n^2 \delta(\mathbf{x}, \mathbf{x}') \,, \tag{3}$$

which describes the covariance between $\mathbf{x}$ and $\mathbf{x}'$ where ℓ controls the "influence range", the larger ℓ the more extensive influence between data and the function δ is the Kronecker delta function. The function is also referred to the Radial Basis Function (RBF) kernel in Support Vector Machines (SVM) [8], [3].

B. Anisotropic Kernel Function

We shall design an appropriate kernel function to describe the relation for each pair of sensor readings given the

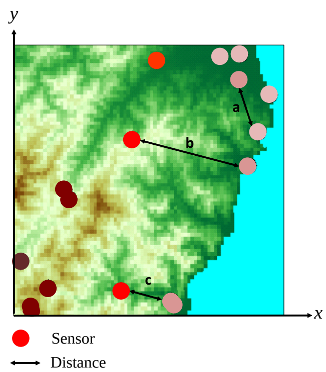


Figure 1. The relationship between proximity and behavior difference. (a) and (b) show that smaller geographic distance implies similar patterns, while (c) shows another case when small distance does not imply small pattern difference.

Gaussian process regression modeling. Various of kernel functions are available for Gaussian Process Regression. For example, we can choose squared exponential, Ornstein-Uhlenbeck, matérn, etc as the candidate kernel function. Among them, squared exponential is a widely used covariance function [2] because the computation is based on a simple metric, the Euclidean distance. Although the kernel function is intuitive and easy to implement, sometimes it can not easily be applied to certain problems. When we have different features where each of them has its own portion of contribution to the regression result, the plain Euclidean-based squared exponential kernel may not be acceptable.

As a kernel function, squared exponential function represents the relation between each pair of sensor readings. If the readings have high similarity, they shall have large covariance or small Euclidean distance. Some issues arise when we can not use a kernel function based on the plain Euclidean distance (such as the kernel function in Eq. 2) between different sensor readings. Let us discuss the relation between different temperature readings, such as the one shown in Fig. 1. We study two cases of relationship between pairs of sensor readings.

The case when isotropic kernel works: In Fig. 1, sensors that are connected by distance **a** have very similar values (shown by the same color). These two sensors also have relatively small Euclidean distance. On the other hand, a pair of sensors that are connected by distance **b** have different sensor colors and they are located far away from each other. If we use Euclidean distance and (x, y) coordinates in 2-D as the features to build a kernel function, we will obtain a covariance function close to what we expect. That is, the covariance between two sensors that share a small Euclidean distance will have similar sensor readings and vice versa.

The case when we need an anisotropic kernel: To study another case, we can focus on a pair of sensors with distance **c** in Fig. 1. Two sensors are located relatively close to each other. If we have $\mathbf{c} < \mathbf{a}$, we expect even more similar readings in the pair of sensors with distance **c** instead of distance **a**. However, it is not the case because the pair of sensors with distance **c** are from two kinds of geographic regions, one is close to the ocean and the other is on the small-hill region and they should not have similar temperature readings. On the other hand, the two sensors with distance **a** should have similar temperature readings because both of them are close to the ocean. If we use squared Euclidean distance and 2-D (x, y) coordinates as the features to build a kernel function for GPR in this case, the result may not be acceptable because the covariance can not well represent the data's relation. An alternative way to solve the puzzle is to add additional features other than (x, y) such as the elevation information z (shown by the color in Fig. 1) and the Euclidean distance on the 3-D space is hopefully well represents the relation between sensor readings. To solve the problem systematically, we look for a general strategy to build a kernel for various of attributes and it needs very little domain knowledge.

Based on the prior knowledge for different sensor data, we know that sometimes certain features may have more contribution to the regression result than others. For example, to predict temperature, latitude information is generally more important than longitude information, and elevation is the most influential feature among all three. Motivated by the above observation, we propose a way to modify the kernel function. In our design, we give a weight for each feature. By estimating appropriate weights for different features, we can determine the portions of contribution for every feature to the regression result. We propose a general kernel function called *anisotropic kernel* as:

$$cov(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}\sum_{i=1}^{d} w_i(\mathbf{x}_i - \mathbf{x}'_i)^2\right) + \sigma_n^2 \delta(\mathbf{x}, \mathbf{x}'), \quad (4)$$

where d is the dimensionality or the number of features we have and $\mathbf{x}_i$, $\mathbf{x}'_i$ are the $i-$features of $\mathbf{x}$ and $\mathbf{x}'$ respectively. In the formula, we use an anisotropic squared exponential covariance function to describe the relation between two sensor readings $\mathbf{x}$ and $\mathbf{x}'$. We choose the sum of the weights to be 1. In the next section, we first introduce a real case study on which we can apply the proposed idea and demonstrate how we can use the proposed idea to solve the real problem: to predict temperature in Taiwan.

III. Case Study

To demonstrate how we can use the proposed method to solve real-world problems, we focus on a temperature prediction given Taiwan's temperature data. Taiwan temperature dataset is an archive of temperature sensor readings collected from regions around the whole Taiwan. We determine to use Taiwan temperature data for our case study because of two considerations. First, the data are relatively easy

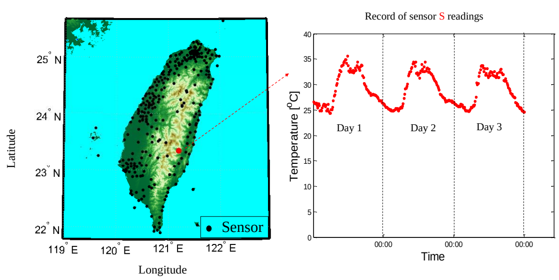


Figure 2. More than 250 temperature sensors are deployed all over the places around Taiwan. Different places may own different sensor density such as fewer sensors deployed in mountainous area rather than in other areas. The right figure shows the temperature readings of a red point sensor on the left figure for several days. Obviously, there is a periodic pattern on the right figure. The temperature readings of next days have almost the same pattern with the current day.

to obtain from public domains. That makes the further comparisons from other methods possible. Second, the data were collected from more than 250 sensors installed in various locations around Taiwan. We can assume that those locations are surrounded with different environmental conditions and sensors on those locations are likely to collect data with high variety. For instance, some of them are deployed in mountainous area while some others are in the area near the sea, and etc. We aim to demonstrate that the proposed method can be effective for prediction given such a high variety dataset. Fig. 2 shows where the temperature sensors were deployed. As we can see, sensors are deployed in various locations, including mountains, hills, tablelands, plains, basins, coastal area, isolated islands, etc. We do have dense sensor deployment in places with large populations.

A. Temperature Sensor Readings

Temperature is one of the most important information for our daily life. Everyday, many people care about weather forecast which includes temperature prediction and many others. To fulfill the need, numerous sensors were deployed around our environment and continue to collect important information for forecast. A common issue includes frequent sensor failure which may need annoying sensor replacement or the replacement may be not possible at all. Another issue that becomes more serious these days is the problem of large amount of energy consumption and bandwidth usage when sending the sensor data. One may want to turn off some sensors for certain period of time for energy saving while the sensor data can still be collected without too much information loss. After all, we look for a robust sensor network where the network continues to collect reliable data even with a small amount of missing data in the network.

B. Sensor Reading Neighborhood

We have to investigate the spatial and temporal relationships between data so that missing data can be recovered based on their spatial or temporal neighbors. Given the spatial relationship between data, we can expect that temperature readings from regions with high proximity are very close to each other. Sometimes we have to consider some regional characteristics such as elevation, close to sea or not, etc, to decide the spatial relationship. On the other hand, with the temporal relationship between data from temporal neighbors, the current temperature value is likely to be close to previous temperature values. We define the spatial and temporal neighborhoods as follows:

- *Spatial neighborhood:* It means the sensor readings collected from nearby geographic locations or with similar characteristics such as elevation.
- *Temporal neighborhood:* It means the readings that belong to similar moments. There is a daily periodicity in temperature data. The current temperature will be close to previous temperatures, and also temperatures at same time but previous days.

Spatial and temporal neighborhood relationships are essential information in temperature prediction. We can use spatial or temporal relationship separately for temperature prediction; on the other hand, we can integrate the two relationships together to improve the prediction. To ease the computation load, we combine temporal information and aggregated spatial information; also, the spatial information and the aggregated temporal information for prediction.

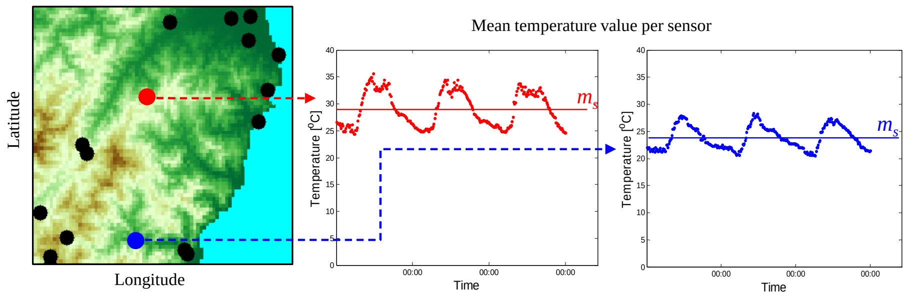
Fixed a moment t, we compute the usual temperature n_t of the moment, called the *aggregated spatial information of the moment*. On the other hand, fixed a sensor s, we compute the usual temperature m_s of the sensor, called the *aggregated temporal information of the sensor*. That is,

$$n_t = \frac{1}{N}\sum_{s=1}^{N} T_t^s\,, \quad m_s = \frac{1}{M}\sum_{t=1}^{M} T_t^s\,, \tag{5}$$

where T_b^a indicates the temperature value on sensor a at time b, and M and N are time interval and number of (regional) sensors respectively.

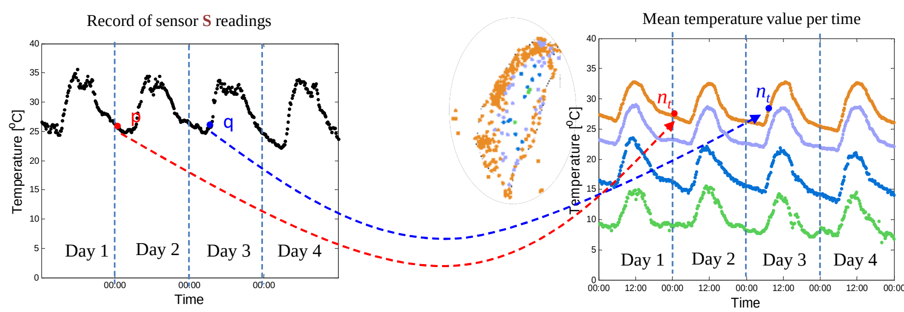
C. Sensor Reading Prediction

We proposed to use spatial and temporal information to predict the temperature on specific sensor for certain time. There are various approaches to do prediction by exploiting the spatial and temporal information in different ways. To predict the temperature of sensor s at time t using the spatial domain information, we can consult all the temperatures $T_t^{N_s(s)}$ at time t for s's spatial neighbors $N_s(s)$, so called the *spatial* approach. In GPR, the data for learning is a set of $(\mathbf{x}, y) = ((\text{latitude}, \text{longitude}, \text{elevation}), \text{temperature})$ for temperature prediction. We denote the spatial approach by $\mathcal{S}$ approach. Similarly, we can define the *temporal* approach $\mathcal{T}$ by collecting the temperature set $T^s_{N_t(t)}$ when we want to predict the temperature of sensor s at time t using the temporal domain information for the temporal neighbors



(a) Spatial information (b) Aggregated temporal information

Figure 3. The approach $\mathcal{S}t$, combining the spatial neighborhood information and aggregated temporal information for prediction. (a) There are two readings from two sensors marked with red and blue color. The relation of these two sensor readings can be defined by sensor's proximity (latitude, longitude) and elevation. (b) Red and blue sensor readings' relation can also be defined by their usual reading values at certain range of time. Usual reading value of a sensor is presented as an aggregated temporal information m_s.



(a) Temporal information (b) Aggregated spatial information

Figure 4. The approach $\mathcal{T}s$, combining temporal neighborhood information and aggregated spatial neighborhood information to do prediction. (a) Two readings from a single sensor but different moments are presented in the left figure, marked as blue and red points. The relation of these two readings can be seen by their time differences (different day, different time) (b) Red and blue sensor readings' relation also can be seen from the usual reading values at those moments. The usual reading value of a certain moment is obtained from aggregated reading values from several nearby sensors (aggregated spatial neighborhood information).

$N_t(t)$ of time t. The learning set for temporal approach $\mathcal{T}$ in GPR is $(\mathbf{x}, y) = (\text{time index}, \text{temperature})$.

To predict the temperature T_t^s using both the spatial and temporal information, we can consider the temperature set $T_{N_t(t)}^{N_s(s)}$ based on both the spatial and temporal neighbors. However, it could be time-consuming if we consider a big set of neighbors for a single-point estimation when we apply GPR. We propose two strategies $\mathcal{S}t$ and $\mathcal{T}s$ to bypass the problem. The first strategy $\mathcal{S}t$ relies more on the spatial information. It integrates the spatial and temporal information by combining the spatial information and aggregated temporal information. The features of a sensor s in this case has an additional aggregated temporal information m_s and the complete features becomes $(s_{\text{lat}}, s_{\text{long}}, s_{\text{elev}}, m_s)$ for learning in GPR. This method can also be illustrated in Fig. 3.

The second approach $\mathcal{T}s$ relies more on the temporal information. It combines the temporal information and aggregated spatial information for regression. The feature set is given by (t, n_t) for a moment t. We also illustrate the idea in Fig. 4. Keep in mind that to have the two methods $\mathcal{S}t$ and $\mathcal{T}s$ work properly for us, we need the help from anisotropic kernel in GPR learning to appropriately balance between different types of features in the set for prediction.

IV. Prediction Result and Analysis

To evaluate how effective the proposed methods is for temperature prediction, we conduct series of experiments to demonstrate that the proposed methods indeed perform well in many occasions. The main purposes of the experiments are to show: (1) considering spatial and temporal information together gives better performance than considering only spatial or temporal information; (2) the anisotropic kernel is more appropriate than isotropic kernel in temperature prediction. In the experiments, we randomly chose twelve sensors, or three sensors for each of the four regions: mountainous area, hills, plains (including coastal area), and isolated area, as the test set to see if the sensor values can be recovered correctly. The twelve selected test sensors and their behaviors are shown in Fig. 5.

The weights for the features in $\mathcal{S}t$ approach were assigned based on prior knowledge; and for the remaining features the weights were decided dynamically. The prior

Table I

THE ABSOLUTE ERROR OF TEMPERATURE PREDICTION BASED ON FOUR METHODS: $\mathcal{S}$, $\mathcal{T}$, $\mathcal{S}t$, AND $\mathcal{T}s$. A BOLD-FACE WITH A STAR INDICATES THE MINIMUM VALUE PER COLUMN. THE FIRST THREE SENSORS BELONG TO MOUNTAINOUS AREA, FOLLOWED BY THREE SENSORS FROM HILLS, THEN THREE SENSORS FROM PLAINS, AND THE LAST THREE ARE FROM ISOLATED AREA.

A. July, 8 2013 06:00 AM

Approach	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
Isotropic Kernel												
$\mathcal{S}$	1.76	1.96	0.66	2.11	2.64	1.34	1.08	2.32	2.63	1.98	0.95	2.46
$\mathcal{T}$	0.30	0.76	**0.03***	0.87	0.03	0.24	0.64	0.39	0.85	**0.27***	0.43	**0.36***
$\mathcal{S}t$	0.62	1.11	0.66	0.80	2.20	0.85	0.34	1.11	1.88	1.64	0.37	1.63
$\mathcal{T}s$	0.39	0.75	**0.03***	0.38	0.10	0.29	0.37	0.19	**0.70***	0.46	0.18	0.47
Anisotropic Kernel												
$\mathcal{S}t$	0.44	**0.63***	0.23	0.38	1.46	**0.18***	**0.13***	0.33	1.87	1.08	**0.03***	0.78
$\mathcal{T}s$	**0.27***	0.70	0.05	**0.15***	**0.01***	0.25	0.16	**0.10***	0.80	0.50	0.14	0.41

B. July, 8 2013 14:00 PM

Approach	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
Isotropic Kernel												
$\mathcal{S}$	2.25	1.42	1.12	3.10	4.30	1.01	**0.37***	0.82	5.03	0.57	0.60	1.04
$\mathcal{T}$	4.59	0.36	0.70	0.30	0.57	1.38	1.45	1.82	5.39	1.26	0.17	1.40
$\mathcal{S}t$	1.24	0.35	0.40	0.99	**0.40***	0.56	0.56	1.18	5.65	1.32	0.68	0.47
$\mathcal{T}s$	4.90	0.20	0.26	0.32	0.59	1.23	1.47	1.73	4.87	1.11	0.21	1.27
Anisotropic Kernel												
$\mathcal{S}t$	**0.69***	**0.02***	0.31	**0.04***	0.57	**0.26***	0.66	**0.67***	4.88	**0.19***	0.48	**0.16***
$\mathcal{T}s$	4.59	0.19	**0.05***	0.16	0.61	1.16	1.19	1.01	**4.20***	0.77	**0.06***	1.37

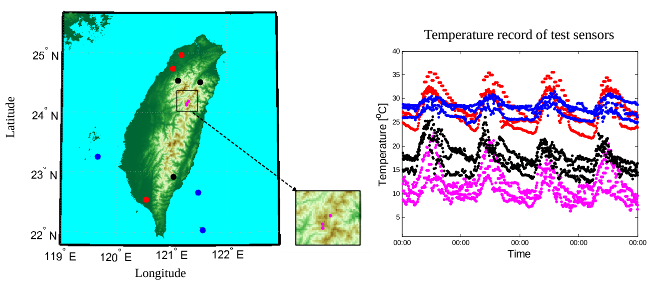


Figure 5. Twelve sensors from different regions are selected as test sensors. There are three sensors in each of the four regions: mountains (magenta), hills (black), plain (red), and isolated area (blue). The right figure shows the temperature records of the twelve selected sensors within several days.

knowledge suggests that the latitude has bigger influence in temperature than the longitude; so the weight for latitude was assigned bigger than the one for longitude. For elevation and aggregated temporal information, the weights were tuned for the best performance. Practically, when we want to predict temperature value T_t^s, we used the latest temperature (T_{t-1}^s) as training data. We try to find the best weights that give minimum error in predicting T_{t-1}^s. Then we applied this weights to predict next temperature T_t^s. We believe that the temperature changes only slightly from $t-1$ to t. So that, the weights for predicting T_{t-1}^s can also be used to predict T_t^s. We use w_a, w_o, w_e, w_n to denote the weight for latitude, longitude, elevation, and aggregated temporal information respectively. Similar to the $\mathcal{S}t$ approach, we also tune the weights for the features in the $\mathcal{T}s$ approach to have the best performance.

The prediction was done by Gaussian Process Regression with the modified kernel function: the anisotropic squared exponential function (Eq. 4). Parameters of GPR were set to be the same for all scenarios. They are $\ell = 10$, $\sigma_f = 1$, $\sigma_n = 0.3$ and the mean function $m(\mathbf{x})$ is set to be 0.

We show the temperature prediction result in Table I. The values that are marked with star(*) are the minimum of absolute error in each column. The absolute error is the value of absolute difference between the real reading value and the prediction result. From Table I, we can see that the integrations of spatial and temporal information ($\mathcal{S}t$, $\mathcal{T}s$) can give better result than those using only spatial or temporal information ($\mathcal{S}$ or $\mathcal{T}$). In addition, the results based on the anisotropic kernel are better than those based on the isotropic kernel. In Table I.A, eight out of twelve sensors had minimum absolute error when we use anisotropic kernel function in GPR. From Table I.B, the experiments with another time, ten of the sensors had minimum absolute error values when using anisotropic kernel function of GPR. Given the test sensors belonging to different locations in Taiwan, GPR with the anisotropic kernel function still can predict temperature value with low absolute error.

A. To choose between Spatial and Temporal information

The previous experiments also show that the integration of spatial and temporal neighborhood information gives better prediction results in most experiments. We want to analyze more about the utilization of those informations such as when we have to use more spatial information than temporal information, and vice versa.

We compare the temperature prediction on sensors in two different regions: plains (or coastal area) and isolated area. We randomly pick three sensors from plains and three sen-

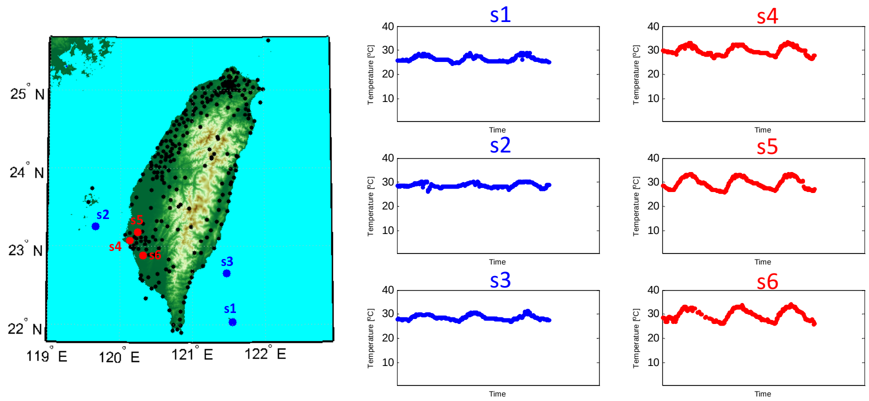


Figure 6. Six sensors were selected as test data. Sensors $S1$, $S2$, and $S3$ are the sensors that belong to isolated area; while sensors $S4$, $S5$, and $S6$ are the sensors that belong to plain or coastal area. The behaviors of those sensors are shown on the right.

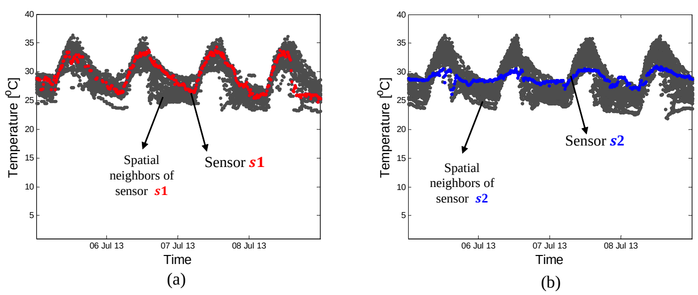


Figure 7. The behavior of sensors and their neighbors. (a) A sensor has similar behavior with its spatial neighbors, (b) a sensor has different behavior with its spatial neighbors.

sors from isolated area. In this case study, there are so many sensors deployed in the plains. It means that the sensors in this area generally have many spatial neighbors. On the other hand, the three sensors from isolated area are the sensors that have very stable temperature readings along the whole day. The temperatures at noon and night are not significantly different. We can say that the sensors in this group have more temporal neighbors. All the six sensors and their patterns are shown in Fig. 6. We use the proposed methods to predict their temperatures within one day (July, 8 2013) and to show how spatial and temporal information play different roles in the prediction. We used Mean Absolute Error (MAE), the average difference between real reading values (T_i) and predictions (P_i), defined by

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |T_i - P_i| \,, \tag{6}$$

to evaluate the proposed methods. The number N is the number of turned off sensors.

The result is presented in Table. II. For sensors deployed in plains, the result shows that the approach $\mathcal{S}t$ works better than approach $\mathcal{T}s$ given the anisotropic kernel. That implies that spatial neighbors' informations are more important than

Table II
PREDICTION RESULT OF SENSORS THAT BELONG TO COASTS AND ISOLATED AREA BASED ON FOUR METHODS: $\mathcal{S}$, $\mathcal{T}$, $\mathcal{S}t$, AND $\mathcal{T}s$.

	Mean Absolute Error					
Approach	**Sensors on plains**			**Sensors on isolated islands**		
Isotropic	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**
$\mathcal{S}$	0.84	1.18	1.06	1.15	2.03	1.96
$\mathcal{T}$	0.85	0.82	1.40	0.42	0.51	0.63
$\mathcal{S}t$	0.67	0.53	0.79	1.15	1.32	1.80
$\mathcal{T}s$	0.69	0.71	1.17	0.37	0.45	0.56
Anisotropic						
$\mathcal{S}t$	**0.50***	**0.40***	**0.70***	0.67	1.14	1.65
$\mathcal{T}s$	0.55	0.56	0.92	**0.32***	**0.35***	**0.50***

temporal neighbors' information. This case happens when we have many spatial neighbors whose behaviors are similar to the test sensor's behavior. On the other hand, for sensors that were deployed in isolated islands, that is when we have more temporal neighbors than spatial neighbors for the test sensors, or the spatial neighbors of the test sensors own very different patterns. We should better use temporal neighbors' information for prediction than using spatial neighbors' information. This behavior difference between a sensor and its neighbors is illustrated in Fig. 7.

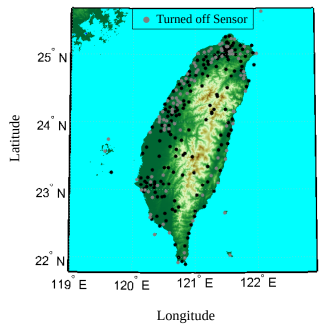


Figure 8. Grey points are the sensors that are put to sleep for few moments.

Table III
PREDICTION RESULT OF TURNED OFF SENSORS BASED ON FOUR METHODS: $\mathcal{S}$, $\mathcal{T}$, $\mathcal{S}t$, AND $\mathcal{T}s$. THE SENSORS WERE TURNED OFF ON JULY, 8 2013.

	Mean Absolute Error (MAE)		
Approach	06:00 AM	14:00 PM	20:00 PM
Isotropic Kernel			
$\mathcal{S}$	1.22	1.26	1.09
$\mathcal{T}$	0.85	1.27	0.86
$\mathcal{S}t$	0.68	1.14	0.89
$\mathcal{T}s$	0.35	1.00	0.86
Anisotropic Kernel			
$\mathcal{S}t$	0.47	0.88	**0.64***
$\mathcal{T}s$	**0.22***	**0.85***	0.70

B. Saving on Power Consumption

We did more experiments on sensor reading prediction to simulate power consumption saving. As we mentioned before (or Fig. 2), there are relatively large number of sensors deployed in plains or coastal area. We can choose to put some sensors deployed in this area to sleep for certain moments. To achieve that, we check whether or not we can predict the slept sensors' readings based on their neighbors' information using our proposed method. The sensors that we turned them off for a while are shown in Fig. 8.

The experiment result in power consumption saving simulation is presented in Table III. The result shows that the proposed methods can perform well for sensor reading prediction in order to decrease power consumption. Given the best methods from the approach $\mathcal{T}s$ or the approach $\mathcal{S}t$, we recover the original readings up to 0.22 to 0.85 MAE (temperature in Celsius).

V. CONCLUSION

Given a set of sensor data, we proposed a mechanism that can save power consumption by using a limited sensor readings to represent the complete sensor readings. The mechanism is based on a study that can find the relation between sensor readings in different locations or collected in different times; that is, we consider the spatial and temporal relations of two sensor readings to decide where and when we can choose not to collect the sensor readings to save energy. We adopt a Gaussian process regressor to recover the missing sensor readings nicely so that the limited awake sensor readings can well represent the other readings from the sensors that are turned to sleep. We proposed various methods that use both spatial and temporal information for prediction and also a strategy that can balance the spatial and temporal relationships of sensor readings in the kernel function for Gaussian process regression. The anisotropic kernel function performed better than the isotropic version. A case study on temperature prediction given Taiwan's weather data showed that the proposed techniques perform well in predicting the turned off sensor readings with low Mean Absolute Error (MAE) given a small set of awake readings. The strategy can be generalized to various attribute set to decide a good kernel function for prediction given a limited learning set.

ACKNOWLEDGEMENT

This research is partially supported by Taiwan National Science Council Grant # 102-2221-E-011-122, also by National Science Council, National Taiwan University and Intel Corporation under grant numbers NSC 102-2911-I-002-001 and NTU 103R7501.

REFERENCES

[1] M.-S. Alouini and A. Goldsmith. Area spectral efficiency of cellular mobile radio systems. *Vehicular Technology, IEEE Transactions on*, 48(4):1047–1066, Jul 1999.

[2] M. Girolami. Mercer kernel-based clustering in feature space. *Neural Networks, IEEE Transactions on*, 13(3):780–784, May 2002.

[3] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[4] H. Jiang, S. Jin, and C. Wang. Prediction or not? an energy-efficient framework for clustering-based data collection in wireless sensor networks. *Parallel and Distributed Systems, IEEE Transactions on*, 22(6):1064–1071, June 2011.

[5] G. Li, Z. Xu, C. Xiong, C. Yang, S. Zhang, Y. Chen, and S. Xu. Energy-efficient wireless communications: tutorial, survey, and open issues. *Wireless Communications, IEEE*, 18(6):28–35, December 2011.

[6] C. E. Rasmussen. Gaussian processes for machine learning. MIT Press, 2006.

[7] D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo. *Advances in Neural Information Processing Systems 8: Proceedings of the 1995 Conference*, volume 8. Mit Press, 1996.

[8] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 2 edition, November 1999.

[9] J. Yan, C. Tian, J. Huang, and Y. Wang. Load forecasting using twin gaussian process model. In *Service Operations and Logistics, and Informatics (SOLI), 2012 IEEE International Conference on*, pages 36–41. IEEE, 2012.