Gaussian Process Regression with Dynamic Active Set and Its Application to Anomaly Detection

Toshikazu Wada [1], Yuki Matsumura [1], Shunji Maeda [2], and Hisae Shibuya [3]
[1] Faculty of Systems Engineering, Wakayama University, 930 Sakaedani, Wakayama, 640-8510 Japan
[2] Hiroshima Institute of Technology, 2-1-1 Miyake, Saeki-ku, Hiroshima, 731-5193 Japan
[3] Yokohama Research Laboratory, Hitachi Ltd., 292 Yoshida-cho, Totsuka-ku, Yokohama, 244-0817 Japan

Abstract - *Gaussian Process Regression (GPR) can be defined as a linear regression in high-dimensional space, where low-dimensional input vectors are projected by a non-linear high-dimensional mapping. Same as other kernel based methods, kernel function is introduced instead of computing the mapping directly. This regression can be regarded as an example based regression by identifying the kernel function with the similarity measure of two vectors. Based on this interpretation, we show that GPR can be accelerated and its memory consumption can be reduced while keeping the accuracy by dynamically forming the active set depending on the given input vector, where active set is the set of examples used for the regression. We call this method Dynamic Active Set (DAS). Based on DAS, we can extend the standard GPR, which estimates a scalar output with variance, to a regression method to estimate multidimensional output with covariance matrix. We applied our method to anomaly detection on real power plant and confirmed that it can detect prefault phenomena four days before actual fault alarm.*

Keywords: Gaussian Process Regression, Example based non-linear regression, Dynamic Active Set, covariance matrix estimation

1 Introduction

Gaussian Process Regression (GPR)[1][2][3] is a well-known non-linear regression method defined as a linear regression in high-dimensional space, where input vectors are projected by a non-linear high-dimensional mapping. Same as other kernel based methods, kernel function is introduced instead of computing the mapping directly.

Unlike the interpretation above, this paper shows another interpretation that GPR can be taken as an example based regression method, where each example consists of two components: input vector and output value. That is, output component of each example is simply weighted by the similarity value between a given input and input vector component of the example, and by summing up them, the output is estimated. Through this interpretation, kernel function is regarded as a similarity function between two vectors. For guaranteeing that input-output relationships in the examples are exactly kept in the regression, a normalization using inverse of gram-matrix is applied.

Based on this notion, we can reduce the size of active set consisting of examples to be used for regression, because only the examples with similar input components with the given input are dominant for output estimation. One contribution of this paper is to form active set dynamically depending on the given input. We call this method Dynamic Active Set (DAS). DAS drastically reduces the computational complexity and the memory consumption of GPR while keeping the accuracy of output.

DAS also breaks the limitation, shared by standard GPR, that only a scalar output and its variance can be estimated. According to the formulae, estimating the vector outputs in the framework of GPR is not a difficult problem. However, the covariance matrix estimation cannot be realized only by simple formula manipulation. Based on the notion above that output value is estimated as a weighted sum of the outputs examples, we propose a method to estimate covariant matrix from the output vectors in the active set with the same weight.

In the following sections, we first show the related works and the interpretation that GPR can be taken as a similarity weighted example based regression. Next, we introduce dynamic active set formation. Then, multivariate extension of GPR is proposed. In the experiments, we applied the resulted method, i.e. DAS based multivariate GPR, to anomaly detection problems and confirmed its efficiency and effectiveness.

2 Related Works

In this section, we first introduce the framework of GPR, and briefly explain some works on improving the computational cost and the memory consumption.

2.1 Gaussian Process Regression

In many literatures, Gaussian Process Regression is explained as a linear regression in a high-dimensional space where input vectors are projected by a non-linear mapping $\boldsymbol{\varphi}(\boldsymbol{x})$.

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{w}$ represents the coefficient vector obeying mean $\mathbf{0}$ isotropic covariance matrix $\sigma^2 I$ Gaussian. That is,

$$\boldsymbol{w} \propto N(\mathbf{0}, \sigma^2 I). \tag{2}$$

Providing N projected inputs: $\Phi = (\boldsymbol{\varphi}(\boldsymbol{x}_1) \quad \cdots \quad \boldsymbol{\varphi}(\boldsymbol{x}_N))^T$, and no information specifying the coefficient vector $\boldsymbol{w}$ is provided, the corresponding outputs: $\boldsymbol{y} = (y_1 \quad \cdots \quad y_N)^T$ can be represented as

$$\boldsymbol{y} = \Phi \boldsymbol{w}. \tag{3}$$

The distribution of $\boldsymbol{y}$ is also a Gaussian as shown below.

$$E[\boldsymbol{y}] = \Phi E[\boldsymbol{w}] = \mathbf{0}, \tag{4}$$

$$cov[\boldsymbol{y}] = E[\boldsymbol{y}\boldsymbol{y}^T] = \Phi E[\boldsymbol{w}\boldsymbol{w}^T]\Phi^T = \sigma^2\Phi\Phi^T = K, \quad (5)$$

where K represents gram matrix consisting of kernel functions between input vectors. That is, a kernel function $k(\boldsymbol{x}_n, \boldsymbol{x}_m)$ represents scalar product $\sigma^2\boldsymbol{\varphi}^T(\boldsymbol{x}_n)\boldsymbol{\varphi}(\boldsymbol{x}_m)$. That is,

$$\boldsymbol{y} \propto N(\boldsymbol{0}, K). \quad (6)$$

When training samples, information on the coefficient vector $\boldsymbol{w}$ is provided, and the estimation will be biased. Providing input-output training data $(\boldsymbol{x}_1, t_1), \cdots, (\boldsymbol{x}_N, t_N)$ consisting of input vector $\boldsymbol{x}_i$ and corresponding output scalar value t_i, the output mean and variance for input $\boldsymbol{x}$ are represented as below.

$$\mu_{GP}(\boldsymbol{x}) = \boldsymbol{k}^T(\boldsymbol{x})K^{-1}\boldsymbol{t}, \quad (7)$$

$$\sigma_{GP}^2(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}^T(\boldsymbol{x})K^{-1}\boldsymbol{k}(\boldsymbol{x}), \quad (8)$$

where $\boldsymbol{t} = (t_1 \quad \cdots \quad t_N)^T$, $\boldsymbol{k}(\boldsymbol{x}) = (k(\boldsymbol{x}_1, \boldsymbol{x}), \ldots, k(\boldsymbol{x}_N, \boldsymbol{x}))^T$, $K = [k(\boldsymbol{x}_n, \boldsymbol{x}_m)]$.

In practice, training data may contain errors like

$$t_n = y_n + \varepsilon_n. \quad (9)$$

Here we assume that the error ε_n is a mean $\boldsymbol{0}$ variance β^2 Gaussian, which is independent of y_n. In this case, we need small modifications: redefine $K = [k(\boldsymbol{x}_n, \boldsymbol{x}_m) + \beta^2]$, and replace Equation (8) by

$$\sigma_{GP}^2(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) + \beta^2 - \boldsymbol{k}^T(\boldsymbol{x})K^{-1}\boldsymbol{k}(\boldsymbol{x}). \quad (10)$$

Same as other kernel based methods, kernel function can be selected from wide varieties of functions satisfying Mercer's condition. One widely used example is the RBF kernel shown below.

$$k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \exp\left(\frac{-\|\boldsymbol{x}_n - \boldsymbol{x}_m\|^2}{\sigma_h^2}\right). \quad (11)$$

2.2 Fast and Memory Efficient GPRs

The dominant computation for the estimation is to compute K^{-1}. Its computational complexity is $O(N^3)$, and the spatial complexity to store the gram matrix K is $O(N^2)$. For the accuracy, the bigger N is the better, but smaller N is preferable for real-time applications.

For solving this problem, the following methods have been proposed [3].

1. Subset of regressors[4][5]: Pick up M examples out of active set consisting of N examples, and use the following approximations.

$$\mu_{SR}(\boldsymbol{x}) = \boldsymbol{k}_M^T(\boldsymbol{x})(K_{NM}K_{MN} + \beta^2 K_{MM})^{-1}K_{MN}\boldsymbol{t}, \quad (12)$$

$$\sigma_{SR}^2(\boldsymbol{x}) = \beta^2\boldsymbol{k}_M^T(\boldsymbol{x})(K_{NM}K_{MN} + \beta^2 K_{MM})^{-1}\boldsymbol{k}_M(\boldsymbol{x}), \quad (13)$$

where K_{NM}, K_{MN}, and K_{MM} represent $M \times N$, $N \times M$, and $M \times M$ gram matrices, respectively. $\boldsymbol{k}_M(\boldsymbol{x})$ represents a vector consisting of kernel functions between $\boldsymbol{x}$ and picked up M input examples.

2. The Nyström Method[6]: Pick up M examples, and approximate gram matrix by

$$\widetilde{K} = K_{NM}K_{MM}^{-1}K_{MN}. \quad (14)$$

3. Subset of Datapoints: Pick up M examples, and simply approximate the gram matrix by K_{MM}.

4. Projected Process Approximation: Pick up M examples, and approximate the mean by equation (12) and variance by

$$\sigma_{PA}^2(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_M^T(\boldsymbol{x})K_{MM}^{-1}\boldsymbol{k}_M(\boldsymbol{x}) + \beta^2\boldsymbol{k}_M^T(\boldsymbol{x})(K_{MN}K_{NM} + \beta^2 K_{MM})^{-1}\boldsymbol{k}_M(\boldsymbol{x}). \quad (15)$$

5. Bayesian Committee Machine[7]: Partition the dataset into p subsets and estimate outputs and variances at multiple test points.
6. Iterative Solution of Linear Systems[8] : An acceleration using iterative conjugate gradient method.

Methods 1,2,3,4 requires the reduction of examples from N to M, which is done by random selection or greedy algorithm described in Algorithm1.

Input: M desired size of active set
Initialization: $\mathcal{D} := \emptyset$, $R = \{1, \cdots, N\}$
for $j := 1$ **to** M
  Create working set $J \subseteq R$
  Compute Δ_j for all $j \in J$
    $i := \arg\max_{j \in J} \Delta_j$
  $\mathcal{D} := \mathcal{D} \cup \{i\}$, $R := R \backslash \{i\}$
endfor
return $\mathcal{D}$

Algorithm1: Greedy algorithm to reduce the size of active set (extracted from [3] and modified.)

The big problem arose here is the computational cost of Δ_j, which represents the gain obtained by adding $\boldsymbol{x}_j$ into the active set $\mathcal{D}$. Foregoing researches propose *differential entropy score*[9], *information gain criterion*[10], as Δ_j. All of their computational costs are expensive, because the measure Δ_j is evaluated over all potential inputs.

Our idea is if the active set $\mathcal{D}$ is dynamically formed depending on a specific input $\boldsymbol{x}$, the measure $\Delta_j(\boldsymbol{x})$ can be more simple and $\mathcal{D}(\boldsymbol{x})$ is easily obtained.

3 GPR with Dynamic Active Set

This section presents our method that reduces the computational cost while keeping the accuracy and extends scalar output to vector output with covariance matrix.

3.1 GPR as a similarity weighted example based regression

$\boldsymbol{k}^T(\boldsymbol{x})K^{-1}$ in Equation (7) can be regarded as a weight vector to the output examples $\boldsymbol{t} = (t_1 \quad \cdots \quad t_N)^T$ (See Fig. 1).

From the viewpoint of similarity, the output for input $\boldsymbol{x}$ can be roughly estimated just by $\boldsymbol{k}^T(\boldsymbol{x})\boldsymbol{t} = \sum_{i=1}^{N} k(\boldsymbol{x}, \boldsymbol{x}_i)t_i$, because of the following facts.

If we regard $k(\boldsymbol{x}, \boldsymbol{y})$ as a similarity measure between $\boldsymbol{x}$ and $\boldsymbol{y}$, we can assume

$$k(\boldsymbol{x}, \boldsymbol{x}) \geq k(\boldsymbol{x}, \boldsymbol{y}). \tag{16}$$

Then the weight $k(\boldsymbol{x}, \boldsymbol{x}_i)$ is maximized at $\boldsymbol{x} = \boldsymbol{x}_i$, i.e., the weight of output example t_i is maximized at $\boldsymbol{x} = \boldsymbol{x}_i$.

However, this formulation does not keep the input-output relationship in the examples. That is, $\boldsymbol{k}^T(\boldsymbol{x}_i)\boldsymbol{t} \neq t_i$, $(i = 1, \cdots, N)$.

For guaranteeing the input-output relationship, the weight vector for the input $\boldsymbol{x}_i$ should be

$$\boldsymbol{\delta}_i = \Big(\underbrace{0 \quad \cdots \quad 0}_{i-1} \quad 1 \quad \underbrace{0 \quad \cdots \quad 0}_{N-i}\Big)^T, \tag{17}$$

because $\boldsymbol{\delta}_i^T \boldsymbol{x}_i = t_i$, $(i = 1, \cdots, N)$.

We can show that $\boldsymbol{k}^T(\boldsymbol{x}_i)K^{-1} = \boldsymbol{\delta}_i^T$ as follows.

For full rank gram matrix K,

$$KK^{-1} = I \tag{18}$$

always stands. By multiplying $\boldsymbol{\delta}_i^T$ with both sides of Equation (13), we get

$$\boldsymbol{\delta}_i^T KK^{-1} = \boldsymbol{k}^T(\boldsymbol{x}_i)K^{-1} = \boldsymbol{\delta}_i^T. \tag{19}$$

For preserving the input-output relationship, $\boldsymbol{k}^T(\boldsymbol{x})K^{-1}$ is the ideal weight vector at least for $\boldsymbol{x}_i$.

Almost the same mathematical formula can be found in the works by S.W. Wegerich[11][12][13]in the context of anomaly detection. This method is called similarity based modeling (SBM). This method is almost the same as GPR except the following properties.

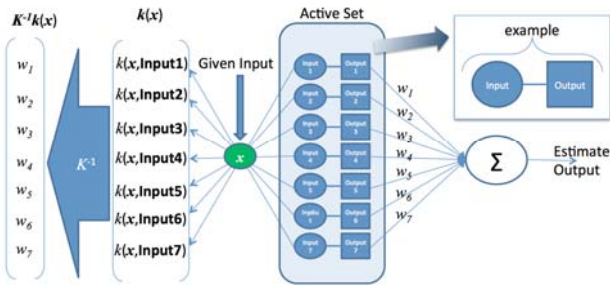


Fig. 1. An interpretation of GPR mean estimation

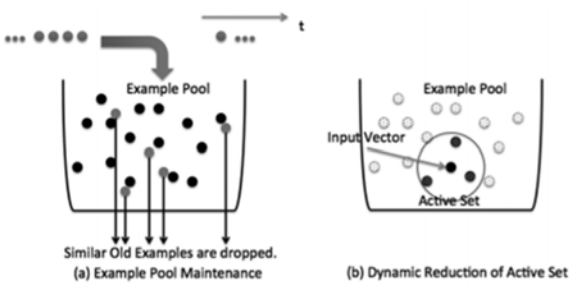


Fig. 2. Example pool and active set formation: (a) Excluding similar examples from the pool (b) Dynamic active set formation

- SBM can estimate vector values, but standard GPR can't.
- GPR can estimate output variance, but SBM can't.
- SBM normalizes the weight vector so that the sum equals to 1, but GPR doesn't.

Our question is whether the kernel function $k(\boldsymbol{x}, \boldsymbol{x}_i)$ can be an importance measure of $\boldsymbol{x}_i$ for estimating the output and variance for $\boldsymbol{x}$ or not. For the input $\boldsymbol{x} = \boldsymbol{x}_i$, the i-th components of $\boldsymbol{k}(\boldsymbol{x})$ and $\boldsymbol{k}^T(\boldsymbol{x})K^{-1}$ are the biggest as shown above. This implies that $k(\boldsymbol{x}, \boldsymbol{x}_i)$ can be an importance measure of $\boldsymbol{x}_i$ when $\boldsymbol{x} \in \{\boldsymbol{x}_j\}$.

The remaining question is: when an input example $\boldsymbol{x}_i$ is the nearest to the given input $\boldsymbol{x}$, still the i-th component of $\boldsymbol{k}^T(\boldsymbol{x})K^{-1}$ is the biggest or not? For answering the question, we introduce the assumption that the kernel function satisfies

$$k(\boldsymbol{x}, \boldsymbol{y}) \geq 0, \tag{20}$$

for any $\boldsymbol{x}$ and $\boldsymbol{y}$. Under this assumption, the components in the vector $\boldsymbol{k}(\boldsymbol{x}) = (k(\boldsymbol{x}, \boldsymbol{x}_1) \quad \cdots \quad k(\boldsymbol{x}, \boldsymbol{x}_N))^T$ dissimilar with $\boldsymbol{x}$ will be close to zero. For such dissimilar input examples $\boldsymbol{x}_i$, the corresponding weight w_i will be closer to zero, where $\boldsymbol{k}^T(\boldsymbol{x})K^{-1} = (w_1 \quad \cdots \quad w_N)$.

As a consequence of above discussion, for kernel functions satisfying Inequalities (16) and (20), it is clear that kernel function can be used as Δ_j. That is,

$$\Delta_j(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}_j). \tag{21}$$

Note that the most distinguishing point from other Δ_j s, Equation (21) has an argument. This implies that the importance of an example cannot be defined apart from the given input $\boldsymbol{x}$.

3.2 Dynamic Active Set

By using Equation (21), we can dynamically select an active set depending on the input $\boldsymbol{x}$ by gathering the examples $\boldsymbol{x}_i$ having bigger $k(\boldsymbol{x}, \boldsymbol{x}_i)$. Suppose that N and M are the sizes of all examples and reduced active set, we have to compute N kernel functions before the reduction and the computational complexity for computing inverse of gram matrix is $O(M^3)$.

The advantage of this method is the computational cost of kernel function is much cheaper than *differential entropy*

score or *information gain criterion*. Further, since $N \ll M^3$ stands in many practical problems, the total computational complexity including active set formation can be approximated by $O(M^3)$.

One thing we have to avoid is to include almost the same examples in the active set. If $\boldsymbol{x}_i = \boldsymbol{x}_j$, $k(\boldsymbol{x}_k, \boldsymbol{x}_j) = k(\boldsymbol{x}_k, \boldsymbol{x}_i)$ stands for all $\boldsymbol{x}_k$ in the active set. This means *i-th* and *j-th* row and columns in the gram matrix are the same, hence the gram matrix is singular and its inverse cannot be obtained. For avoiding this case, we introduce example pool that excludes almost the similar example.

For time series data, new examples are sequentially injected to the pool. When the kernel function between the injected data and an example in the pool exceeds the given threshold, the example in the pool is dropped and the injected data is stored in the pool as shown in Fig. 2. This pooling mechanism is intended to refer newer examples for representing recent trend.

The above pooling mechanism is an example design, but the most important function of the example pool is to exclude the similar data for stable computation of the gram matrix inverse.

3.3 Multivariate GPR

The extension of GPR to estimate vector output is very simple. By replacing the output example vector $\boldsymbol{t} = (t_1 \quad \cdots \quad t_N)^T$ in Equation (7) or (12) by matrix consisting of vector output examples $T = (\boldsymbol{t}_1 \quad \cdots \quad \boldsymbol{t}_N)^T$, the expected vector output can be estimated. However, in this case, we have to estimate the covariance matrix. Unfortunately, Equation (8), (10), (13), or (15) cannot simply be extended to estimate covariance matrix. The essential difficulty lies in estimating the covariance among the outputs.

The advantage of our method DAS is that we can reduce the input-output examples depending on the given input $\boldsymbol{x}$ and their weight vectors are computed as $\boldsymbol{k}^T(\boldsymbol{x})K^{-1} = (w_1 \quad \cdots \quad w_M)$. These fact implies a simple covariance matrix estimation: Suppose that $(\boldsymbol{x}_1 \quad \cdots \quad \boldsymbol{x}_M)$, $(\boldsymbol{t}_1 \quad \cdots \quad \boldsymbol{t}_M)$, $(w_1 \quad \cdots \quad w_M)$ are the reduced input examples, output examples, and weight values for given input $\boldsymbol{x}$. Then the output $\boldsymbol{\mu}$ and its covariance matrix Σ can be estimated as

$$\boldsymbol{\mu} = \frac{1}{\sum_{i=1}^{M} w_i} \sum_{i=1}^{M} w_i \boldsymbol{t}_i, \tag{22}$$

$$\Sigma = \frac{1}{\sum_{i=1}^{M} w_i} \sum_{i=1}^{M} w_i (\boldsymbol{t}_i - \boldsymbol{\mu})(\boldsymbol{t}_i - \boldsymbol{\mu})^T. \tag{23}$$

For those inputs same with one of the stored input components $\boldsymbol{x}_i$, $(i = 1, \cdots, M)$, $\sum_{i=1}^{M} w_i = 1$ automatically stands, because the weight vector will be $\boldsymbol{\delta}_i$. This means that the Equation (22) is essentially equivalent to Equation (7). This implies that the above equations are not far from the principle of GPR.

From these equations, since we can estimate the output vector and its covariance matrix, we can measure the Mahalanobis distance of the observed output from the expected output. This can be an anomaly measure of a system.

4 Experiments

In this section, we first show how DAS improves computational time while keeping the accuracy. Next, we examine the validity of the vector output and covariance estimation property by using 2D swiss roll data. Finally, our method is applied to an anomaly detection problem of a power plant, which is practically used in real world and stopped because of a fault. Among the sensor values attached to this plant, we picked up two sensor values and compared the sensitivities of the Mahalanobis distances for independent sensors and simultaneous analysis as 2D sensor values.

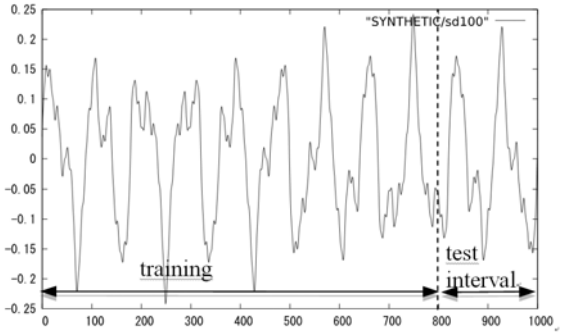


Fig. 3. Training and test intervals assigned on the artificial data [14]

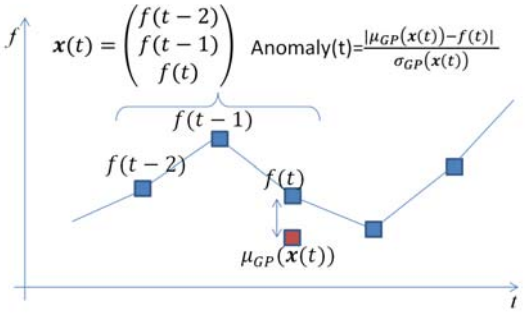


Fig. 4. Input-output assignment and anomaly measure.

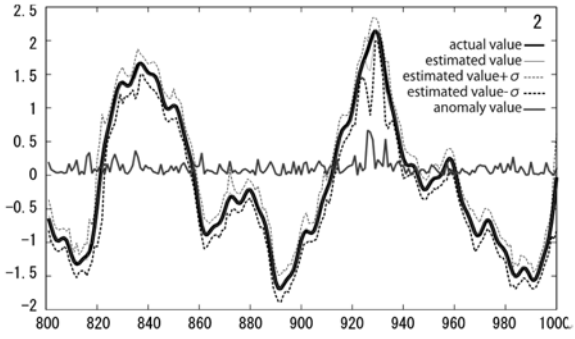


Fig. 5. Actual value, estimated value, estimated value ±σ, and anomaly value in test interval (Active set size = 2).

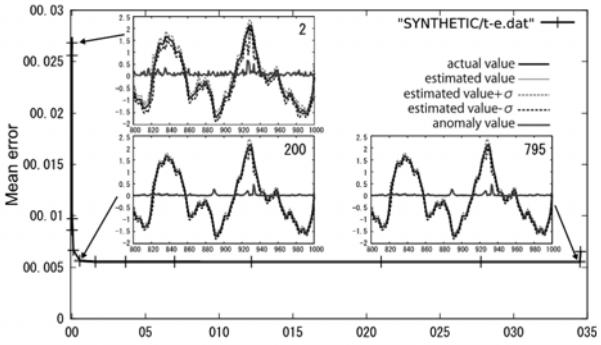


Fig. 6. Computational time V.S. mean absolute error

4.1 Computational Time and Accuracy

In this experiment, we apply DAS based GPR to anomaly detection problem of a temporal sequence $f(t)$. The purpose is to evaluate the relationship between the estimation error and estimation time.

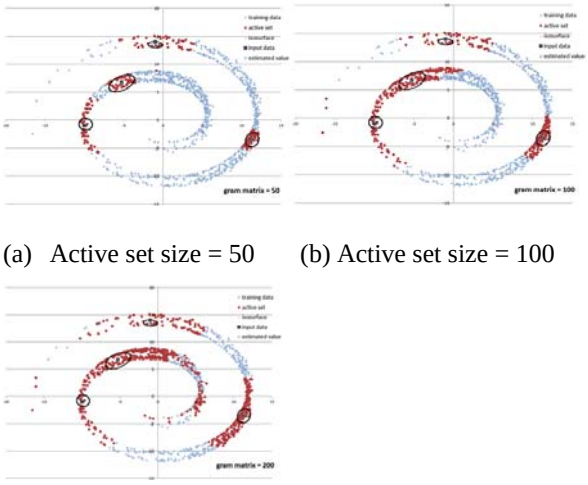
The input vector is $\boldsymbol{x}(t) = (f(t-2) \quad f(t-1) \quad f(t))^T$ and the output is $f(t)$. By dividing the absolute difference between the estimated mean $\mu_{GP}(\boldsymbol{x}(t))$ and $f(t)$ by the estimated standard deviation $\sigma_{GP}(\boldsymbol{x}(t))$, we obtain the anomaly measure.

The sequence data is an artificially generated that was used in waveform retrieval research[14]. The original data consists of 10000 data points. In this experiment, we resample the data to 1000 points and first 800 points are used for training data and last 200 points are used for test. In this experiment, we used RBF kernel with $\sigma_h^2 = 0.1$ and noise $\beta^2 = 0.01$, and we didn't use example pool. The computer is Core2 Duo 1.86 GHz, and the GPR is implemented as a single thread program by C language.

Fig.5 shows an example of estimation in test interval at active set size is only 2. Even at this poor setting, actual value is within the estimated value $\pm\sigma$. The mean absolute error in this interval is 0.0268, which is already small.

Finally, we applied our multivariate GPR to a real power plant data. In this experiment, we used two sensor data. Both are sampled every 30 seconds. We take these sensor data as a temporal sequence of 2D vector. The power plant is activated every morning and stopped every evening. Because of this human intervention, the sensor data behaves nonlinearly. As shown in Figure 3, we confine ourselves to use sensor data sampled at $t-2$, $t-1$, and t for estimating the sensor value at t. So, if we use 2 sensor data, the estimation will be a regression from 6D vector to 2D vector as shown in Figure 8. Also, we can perform 3D to 1D and 6D to 1D regressions.

We performed all these regressions and measured the Mahalanobis distance from the estimated mean, providing one month data (October) as training data and the test interval is November 1-10, where embedded alarm system detected the fault during November 8-10.



(a) Active set size = 50 (b) Active set size = 100

(c) Active set size = 200

Fig. 7. From 2D to 2D regression results on swiss roll data with Mahalanobis distance = 9 ellipses. Red brown points represent active sets

The results are shown in Figures 9-11. According to these results, prefault phenomenon seems occured from November Fig. 7 shows the results of ellipses whose Mahalanobis distances are all 9, which means . The active set size is changed 50, 100, and 200. From these plots, we can confirm that the ellipses fit the local point distributions representing local covariance, and the ellipses are almost insensitive to the active set size.

4 to 7, four days earlier than the actual alarm. Compared with the 3D→1D and 6D→1D results, 6D→2D regression provides us the clearest result.

One may think that Figure 9 (b) captures the prefault phenomenon like Figure 11. However, other 3D→1D and 6D→1D regressions are not congruent with each other. This means that only by Figure 9 (b), one cannot conclude that prefault is detected during November 4-7. On the other hand, Figure 11 is an integrated result of two sequences, and the Mahalanobis distance becomes bigger from November 4. Then one can notice something unusual phenomenon happening. These facts supports the superiority of our multivariate regression and anomaly detection.

By increasing the active set size, the computational time may increase but the mean absolute error will decrease. Fig. 6 shows the result.

This "L" shaped plot shows that the mean error is saturated at active set size greater than 200. It means we can accelerate the estimation speed almost 65 times faster in this case while keeping the accuracy. This is the effectiveness of DAS.

In this plot, we can also find that the mean absolute error increases at active set size bigger than 795. This is because the singularity of gram matrix caused by similar example inclusion.

4.2 Multivariate Regression

For testing the validity of our multivariate regression method defined in Equation (22) and (23), here we show some simple regression result.

In this experiment, we use 2D swiss-roll data and the input and the output examples are assigned to the same 2D data. We used RBF kernel with $\sigma_h^2 = 0.1$ and noise $\beta^2 = 0.01$. The data points are sequentially added to the example pool and those data points in the pool having kernel function greater than the threshold 0.998 are excluded from the pool.

The purpose of this experiment is to draw equi-Mahalanobis distance ellipses while changing the size of active set to verify 1) the ellipses represent the local distribution, 2) the shape and position of the ellipse are insensitive to the active set size.

4.3 Anomaly Detection on a Power Plant Data

Finally, we applied our multivariate GPR to a real power plant data. In this experiment, we used two sensor data. Both are sampled every 30 seconds. We take these sensor data as a temporal sequence of 2D vector. The power plant is activated every morning and stopped every evening. Because of this human intervention, the sensor data behaves nonlinearly. As shown in Figure 3, we confine ourselves to use sensor data sampled at $t-2$, $t-1$, and t for estimating the sensor value at t. So, if we use 2 sensor data, the estimation will be a regression from 6D vector to 2D vector as shown in Figure 8. Also, we can perform 3D to 1D and 6D to 1D regressions.

We performed all these regressions and measured the Mahalanobis distance from the estimated mean, providing one month data (October) as training data and the test interval is November 1-10, where embedded alarm system detected the fault during November 8-10.

The results are shown in Figures 9-11. According to these results, prefault phenomenon seems occured from November 4 to 7, four days earlier than the actual alarm. Compared with the 3D→1D and 6D→1D results, 6D→2D regression provides us the clearest result.

One may think that Figure 9 (b) captures the prefault phenomenon like Figure 11. However, other 3D→1D and 6D→1D regressions are not congruent with each other. This means that only by Figure 9 (b), one cannot conclude that prefault is detected during November 4-7. On the other hand, Figure 11 is an integrated result of two sequences, and the Mahalanobis distance becomes bigger from November 4. Then one can notice something unusual phenomenon happening. These facts supports the superiority of our multivariate regression and anomaly detection.

5 Conclusions

In this paper, we first show an interpretation that GPR is a similarity-weighted example based regression. Based on this interpretation, we next propose a computationally effective GPR with dynamic active set (DAS), which forms the active set depending on given input.

DAS is useful not only for the computational effectiveness but also for covariance estimation when estimating vector output. In the experiments, we have shown the following facts.

- DAS accelerates the GPR and reduces the memory use drastically while keeping the accuracy.
- DAS based multivariate GPR can estimate the local distributions around the estimated mean.
- DAS based multivariate GPR drastically improves the sensitivity of the anomaly measure, i.e., Mahalanobis distance from the estimated mean value.

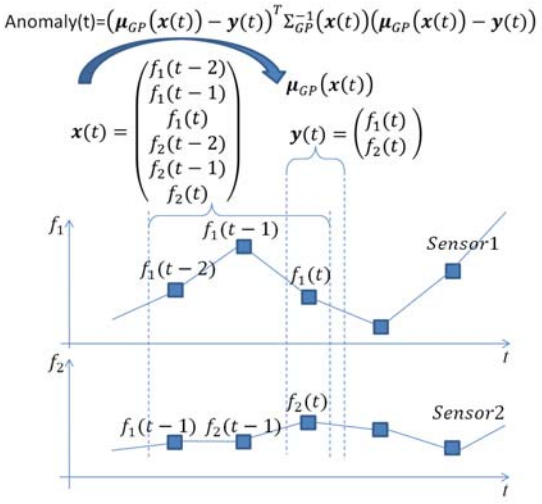


Fig. 8. Anomaly detection scheme for multiple sensor sequences.

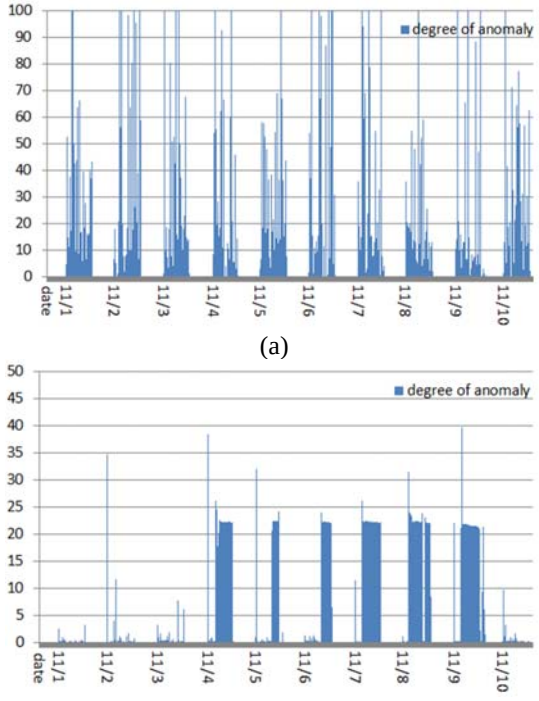


Fig. 9. Mahalanobis distance for sensor 1 obtained by(a)3D→1D regression (b) 6D→1D regression

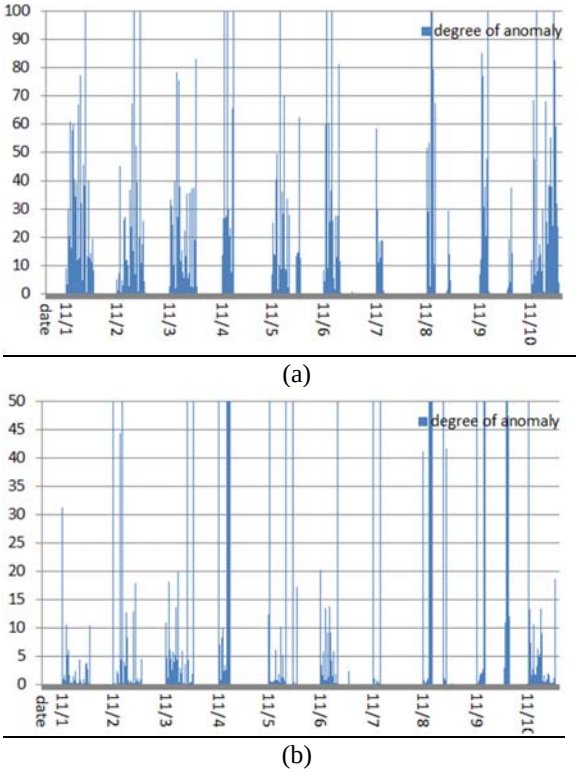


Fig. 10. Mahalanobis distance for sensor 2 obtained by(a)3D→1D regression (b) 6D→1D regression

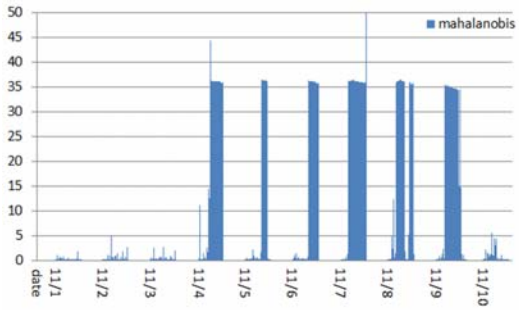


Fig. 11. Mahalanobis distance obtained by 6D→2D regression for both sensors.

Since current implementation of GPR employs example pool that excludes similar examples, our system ignores the example density. This means that our system cannot take account of a priori distribution of the input-output examples. This should be improved in the future works.

6 References

[1] D.J.C. MacKay, "Introduction to Gaussian processes," C.M. Bishop, ed., *Neural Networks and Machine Learning*, volume 168 of NATO ASI Series, pp.133-165, Springer, Berlin, 1998.

[2] C.M. Bishop, " Pattern Recognition And Machine Learning, " Springer-Verlag, Berlin, 2006

[3] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, The MIT Press, Cambridge, MA, 2006.

[4] G. Wahba, "Spline Models for Observational Data," Society for Industrial and Applied Mathematics, Philadelphia, PA. CBMS-NSF Regional Conference series in applied mathematics, 1990

[5] T. Poggio, and F. Girosi, "Networks for Approximation and Learning," *Proceedings of IEEE*, Vol. 78, Issue 9, pp. 1481–1497, 1990

[6] C. K. I. Williams and M. Seeger, "Using the Nyström Method to Speed Up Kernel Machines," In Advances in Neural Information Processing Systems 13, eds. T. K. Leen, T. G. Diettrich, and V. Tresp, pp. 682–688. MIT Press. 2001

[7] V. Tresp, "A Bayesian Committee Machine," *Neural Computation*, Vol. 12, No. 11, pp. 2719–2741, 2000

[8] G. Wahba, D. R. Johnson, F. Gao, and J. Gong, "Adaptive Tuning of Numerical Weather Prediction Models: Randomized GCV in Three-and Four-Dimensional Data Assimilation," *Monthly Weather Review*, 123, pp. 3358–3369, 1995

[9] N. Lawrence, M. Seeger, and R. Herbrich, "Fast Sparse Gaussian Process Methods: The Informative Vector Machine," In Advances in Neural Information Processing Systems 15, eds. S. Becker, S. Thrun, and K. Obermayer, pp. 625–632. MIT Press, 2003

[10] M. Seeger, C. K. I. Williams, and N. Lawrence, "Fast Forward Selection to Speed Up Sparse Gaussian Process Regression," In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*. Society for Artificial Intelligence and Statistics, 2003

[11] S.W. Wegerich, "Similarity based modeling of time synchronous averaged vibration signals for machinery health monitoring," *Proceedings of IEEE Aerospace Conference*, vol.6, no., pp. 3654- 3662 Vol.6, 6-13 March 2004.

[12] S.W. Wegerich, "Similarity-based modeling of vibration features for fault detection and identification", *Sensor Review*, Vol. 25 Iss: 2, pp.114-122, 2005.

[13] S.W. Wegerich, D.R. Bell, and X. Xu, "Adaptive modeling of changed states in predictive condition monitoring", US Pat. 7,233,886 - Filed 27 Feb 2001.

[14] E.J. Keogh, M.J. Pazzani, "An indexing scheme for similarity search in large time series databases," *Proceedings of the 11th International Conference on Scientific and Statistical Database Management (SSDBM)*, Cleveland, Ohio, 1999.