Support Vector Methods and Use of Hidden Variables for Power Plant Monitoring

Chao Yuan[1], Claus Neubauer[1], Zehra Cataltepe[1] and Hans-Gerd Brummel[2]
[1]Siemens Corporate Research, Princeton, New Jersey
[2]Siemens Westinghouse Power Corporation, Orlando, Florida

Abstract

This paper has three contributions to the fields of power plant monitoring. First, we differentiate out-of-range detection from fault detection. An out-of-range refers to a normal operating range of a power plant unseen in the training data. In the case of an out-of-range, instead of producing a fault alarm, the system should notify the operator to include more training data which capture this new operating range. Second, we apply support vector one-class classifier to out-of-range detection for its good volume modeling ability. Third, we propose to use hidden variables in regression models for fault detection. This is shown to be much better than prior work in terms of spillover reduction.

1. Introduction

The task of monitoring a power plant is to detect faults at an early stage and avoid damages to the major components of the plant e.g. gas turbine, steam turbine, generator. In the following, we will concentrate on gas turbines, which are widely used in power generation and which need a high amount of attention during operation. But the presented approach is not restricted to power plant equipment and it can also be applied in other industrial or technical areas, where operation data are available. For gas turbines, early fault detection is typically achieved by analyzing a set of sensors, installed in different parts of the engine measuring the output and additional temperatures, flows and pressures at critical locations. When the turbine is working properly, the sensor data should be distributed in a *normal operating range*. If the sensor data deviate much from this range, there may be a fault (such as a crack in one of the transition pieces of a gas turbine) and a fault alarm should be made. Various statistical models including neural networks, fuzzy logic, independent component analysis and etc have been proposed to learn this normal operating range from training data [1-3].

However, not every deviation is due to a real fault; it can be due to *another normal operating range*, which is not seen in the training data. We refer to this kind of deviation as an *out-of-range*. An out-of-range should be treated differently from a fault, since the machine is still operating normally. More historical data that capture this new operating range should be used to retrain the model to guarantee its accuracy. A good power plant monitoring system should be able to handle both out-of-range detection and fault detection. However, most prior work has ignored the out-of-range problem.

We differentiate two sets of sensors. We refer to the first set as *process drivers*, such as fuel flow and inlet temperature sensors, which represent the inputs of a gas turbine. We refer to the second set as *dependent sensors*, such as power, blade path temperatures, pressures and vibration sensors, which represent internal data and the outputs of the engine. We use process drivers for the out-of-range detection, since they determine the operating state of the plant. Process drivers are relatively independent to each other. Thus, the distribution of the process drivers is more like a volume than a surface. We apply support vector representation machine (SVRM) to this problem for its excellent capability for volume modeling [4, 5].

Fault detection is only applied to dependent sensors, since they measure the performance of the plant. In general, fault detection consists of two steps: sensor estimation and decision [1]. In the sensor estimation step, correct sensor values are estimated; the residues (differences) between the observed values and estimated values are calculated. In the decision step, if the residues are statistically different from zero, the corresponding sensors are marked as faulty. Sensor estimation is our major concern and it must be accurate. Specifically, a faulty sensor's residue should be close to its real deviation and a normal sensor's residue should be close to zero. A common undesired phenomenon for a statistical model is that a normal sensor's residue is affected by a faulty sensor's residue such that both are not close to their ideal values. This is referred to as a spillover problem [3] and should be avoided.

Based on the input-output view of a gas turbine noted above, a straightforward solution for sensor estimation is to use process drivers to estimate dependent sensors. This kind of estimation is also referred to as regression [1]. However, typically, only part of all process drivers are known and many others are unavailable. For example, air humidity is also a process driver, but it is not measured in many circumstances. Thus, only using this partial input information, a regression model will not be accurate.

A solution for this problem is to use the correlation between dependent sensors. For example, in a combustion turbine engine, all blade path temperature sensors are highly correlated such that one sensor could be used to predict the value of another sensor. Inferential sensing

0-7803-8874-7/05/$20.00 ©2005 IEEE

which uses such correlation information for sensor estimation has been proposed [6]. Autoassociative neural networks, kernel regression and its variations: multivariate state estimation techniques (MSET) and the support vector regression (SVR) models have been employed based on this idea [6]. In all these regression models, one correlated sensor is used as the output of the regression model and all the other correlated sensors are used as the inputs. Spillover exists if any of the input correlated sensors is faulty. In this paper, we propose a novel method to reduce spillover in prior regression models. We replace all the input correlated sensors by a single hidden variable, estimated from these correlated sensors. Based on the output estimates, we re-calculate the hidden variable to reduce the spillover effect due to possible faulty sensors.

This paper is organized as follows. In Sect.2, we address using support vector representation machine for out-of-range detection. In Sect.3, we describe the use of hidden variables in regression models for fault detection. In Sect.4, we present our test results. We summarize this paper in Sect.5.

2. SVRM for Out-of-Range Detection

Out-of-range detection is essentially a one-class classification problem, which discriminates between the in-range class and the out-of-range class. No sensor estimation is necessary. Let $\mathbf{y}$ denote the sensor vector consisting of M process drivers. As noted in Sect.1, process drivers are relatively independent to each other (i.e. they can vary independently without being affected by each other). Thus, the distribution of $\mathbf{y}$ is more like a volume than a surface or a curve. We thus apply support vector representation machine (SVRM) [5] to model the distribution of $\mathbf{y}$ for its good ability for volume modeling. Suppose that we are given L training vectors $\{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_L\}$ from the in-range class. The training task is to find an evaluation function $f(\mathbf{y})$, which gives the confidence of the input $\mathbf{y}$ being in the in-range class. We define the decision region $R=\{\mathbf{y}: f(\mathbf{y}) \geq T\}$ to contain those samples $\mathbf{y}$ giving evaluation function values above some threshold T. To achieve a high recognition rate, training vectors should produce high evaluation function values.

We borrow the kernel method used in support vector machines, which defines a mapping Φ from the input space to the feature space [7]. The explicit form of Φ is not necessary. Rather, only the inner product $\Phi(\mathbf{y}_i)^T \Phi(\mathbf{y}_j)$ need be specified to be some kernel function. We consider only the Gaussian kernel $\exp(-\|\mathbf{y}_i-\mathbf{y}_j\|^2/2\sigma^2)$, since it simplifies volume estimation and has other desirable properties. The evaluation function has an inner product form:

$$f(\mathbf{y})=\mathbf{h}^T\Phi(\mathbf{y}), \quad (1)$$

where the solution $\mathbf{h}$ for our SVRM satisfies

$$\left.\begin{array}{c} \text{Min} \|\mathbf{h}\|^2/2 \\ \mathbf{h}^T\Phi(\mathbf{y}_i) \geq T=1,\ i=1,2,...,L \end{array}\right\}. \quad (2)$$

The second condition in (2) insures large evaluation function values for the training set greater than some threshold T. We minimize the norm $\|\mathbf{h}\|$ of $\mathbf{h}$ in the first condition in (2) to reduce the volume of R (to provide rejection of out-of-range samples). In (2), we minimize the square of $\|\mathbf{h}\|$, since such optimization is easily achieved using quadratic programming. Slack variables are also introduced to address outliers in training data. The details of the SVRM can be found in [5]. Using the SVRM, the out-of-range detection is performed as follows. To classify an input $\mathbf{y}$, we compute the evaluation function $f(\mathbf{y})$ in (1); if this is $\geq T=1$, we classify $\mathbf{y}$ as in-range; otherwise, it is out-of-range. In case of an out-of-range, the operator is notified to include more historical data which cover this new range.

3. Use of Hidden Variables in Regression Models for Fault Detection

In this section, we address the fault detection problem, which requires sensor estimation of each dependent sensor. We present a new strategy, which can be combined with any regression model and help to reduce spillover. Instead of inputting multiple correlated dependent sensors (which may contain a faulty sensor) to a regression model, we compute a hidden variable from these correlated sensors and only input this hidden variable to the regression model. A faulty sensor contributes little to this hidden variable and thus shows little effect on the output.

We now detail our method. We divide all dependent sensors into several groups; within each group, all the sensors are highly correlated. This process can be done either using domain knowledge or correlation analysis techniques (see chapter 15 in [2]). We only use one such group as an example. Suppose that there are N correlated sensors in this group, with their values denoted by x_1, x_2, …, x_N. Since they are highly correlated, we define a hidden variable t such that $x_1=g_1(t)+\varepsilon_1$, $x_2=g_2(t)+\varepsilon_2$, …, $x_N=g_N(t)+\varepsilon_N$, where $g_i(t)$ is a *sensor function* defined on t. t can be viewed as an unknown process driver to the power plant. ε_i is the sum of the modeling error (which could be attributed to other known or unknown process drivers) and noise. Since x_i are highly correlated, ε_i is small and is omitted for the rest of this paper for simplicity. If we know t, we can use t as a new input to a regression model to provide more information in sensor estimation. We assume that g_i is invertible such that $t=g_i^{-1}(x_i)$. For all the cases we consider, this is satisfied. If this is not satisfied, we divide t into several segments such that in each segment the sensor function g_i is invertible.

We now present two ways to compute t, based on x_1, x_2, …, x_N. In the first approach, we define simple parametric forms of g_i. For example, $x_i=g_i(t)=a_it+b_i$, where a_i and b_i are the scaling and dc components for the ith sensor, respectively. This is the approach we use and it suffices for all the cases we meet. All the parameters a_i and b_i are computed using least square methods from the training

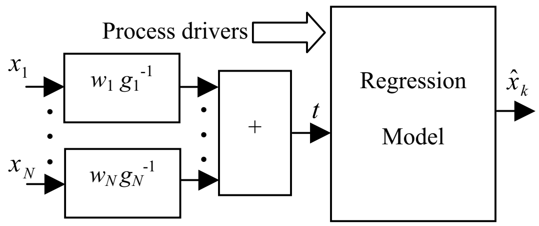

Fig.1. System structure of our regression model

data. In the second approach, to cope with more complicated correlation relationship, we consider nonparametric forms of g_i and can apply principal curves or their variations [8] to obtain g_i. For both approaches, we can simplify the problem by making $t = x_j$, which can be any of x_i. Thus, $x_i = g_i(t) = g_i(x_j)$, where i=1,2,…, N.

Using either of the above two approaches, we can get an estimate of t from each x_i using g_i^{-1}. To reduce the effects of modeling error and noise, we average all these estimates:

$$t = \sum_{i=1}^{N} w_i g_i^{-1}(x_i)\,, \tag{3}$$

where w_i is the confidence weight for each t estimate and $\sum_{i=1}^{N} w_i = 1$. Fig.1 shows the system structure of our proposed regression model. $\hat{x}_k$ is the estimate for the kth correlated sensor. The inputs consist of all the process drivers and the hidden variable t. This is very different from prior work, which directly used x_1, x_2, …, x_{k-1}, x_{k+1}, …, x_N as the inputs to the regression model (in fact, it is unknown if the process drivers were used as inputs [6]). The robustness of a prior work model is questionable, since if any of x_i is faulty, $\hat{x}_k$ will be affected. We handle this spillover problem by assigning different confidence weight w_i for different correlated sensors such that if the ith sensor is likely to be faulty (with a large residue), the corresponding w_i is low. We update w_i based on the residue between the ith sensor and its estimate:

$$w_i = \frac{q(|\hat{x}_i - x_i|)}{\sum_{k=1}^{N} q(|\hat{x}_k - x_k|)}\,, \tag{4}$$

where $q(d)$ is an decreasing error function defined between 0 and +∞ and it produces a value between 0 and 1. We choose a Gaussian function for $q(d)$ such that $q(100)$ = a very small number (0.00001).

We now detail the steps of our sensor estimation algorithm. During training, we first compute g_i^{-1} for all correlated dependent sensors using the training data. Then, we train each regression model in Fig.1 with w_i=1/N, where i=1,2,…, N. During monitoring, we use our model in Fig.1 to compute the estimate for each correlated dependent sensor. Then, we update w_i using (4). If the estimates do not vary much from those obtained in the previous iteration, we output the estimates; otherwise, we repeat the above procedure.

Several points need to be noted. First, the number of regression models is equal to the number of dependent sensors, with one regression model (Fig.1) for one dependent sensor. Second, we assume that only a small number of sensors in each correlation group can be faulty. If most of them are faulty, the calculation of the hidden variable t in (3) is not accurate. However, this is seldom the case. If the number of sensors in one correlation group is small (< 4), we do not calculate t for this group. If there are K correlation groups, each containing a considerable number (≥ 4) of sensors. The inputs for our regression model consists of all the process drivers and K hidden variables (one for each group).

4. Test Results

Due to the limited space, we only present results for fault detection. We use data from a gas turbine of a European combined cycle power plant. A total of 19 sensors are used, including three process drivers: gas flow, inlet temperature, inlet guide vane (IGV) actuator position, and 16 dependent blade path temperature sensors: BPTC1A, BPTC2A, ... and BPTC16A. These 16 BPTC sensors are known to be highly correlated. The task is to estimate the values of these 16 BPTC sensors from the observed values of all 19 sensors. There are a total of 1248 data points. We use the first 600 data points in training and the remaining 648 data points in testing.

We could use real fault cases to test different algorithms. However, the correct value of a sensor is unknown and thus it is difficult to evaluate the accuracy of sensor estimation. We thus consider artificial faults. We add +60 degree step to BPTC1A, between data points 900 and 1248. Figs.2a and b show the faulty sensor and a normal sensor BPTC9A, respectively.

We consider support vector regression (SVR) model [7, 9] in our tests, although any regression model can be tested here. For the prior work method [6], we used three process drivers and 15 BPTC sensors as the inputs of the SVR; the output of the SVR is the estimate of the other BPTC sensor. Using our new system in Fig.1, the inputs of the SVR are three process drivers and the hidden variable t (computed from all 16 BPTC sensors); the output of the SVR is the estimate of the other BPTC sensor. Note that for each method there are a total of 16 regression models, one for each BPTC sensor. We use parametric sensor function $x_i = g_i(t) = a_i t + b_i$ as noted in Sect.3. Figs.2c and d show the residues for the normal sensor BPTC9A using prior work method and our method, respectively. Spillover is clearly seen in Fig.2c while not noticeable in Fig.2d.

To test how different algorithms respond to different scales of the step faults, we vary the step fault magnitude from 0 to 100 in increments of 20 and repeat the above tests. We consider two errors: the *spillover error* (the average absolute residue in the affected period for all 15 normal BPTC sensors) and the *sensor estimation error* (the average absolute difference between the estimate and the

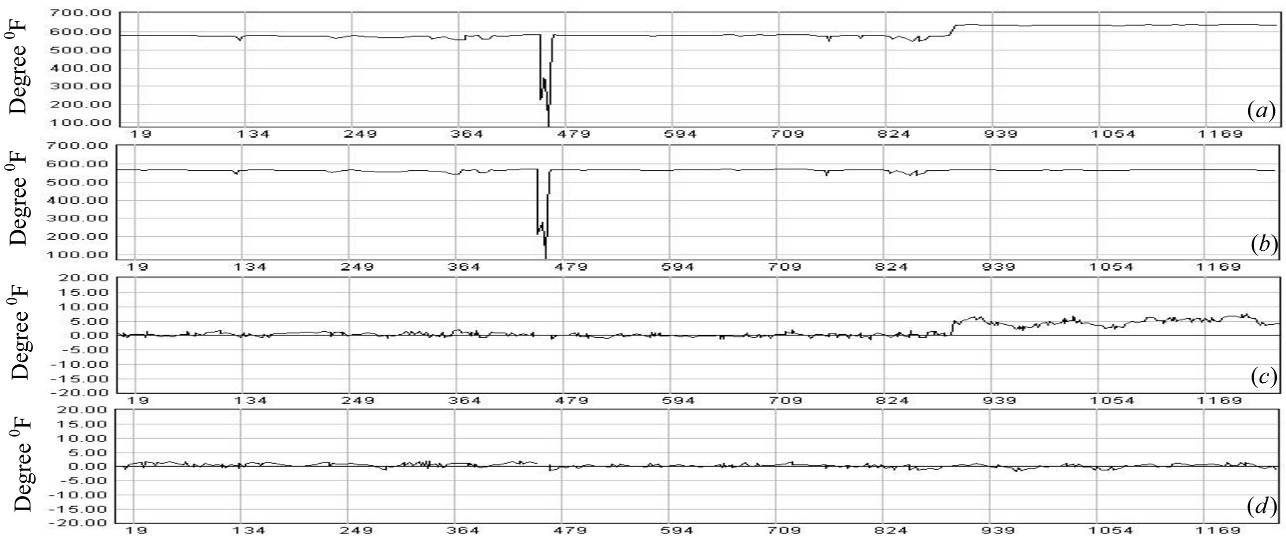

Fig.2. Test results for prior work method and the proposed method. (a) faulty sensor BPTC1A with a +60 step between data points 900 and 1248. (b) normal sensor BPTC9A. (c) residue of BPTC9A using prior work method. (d) residue of BPTC9A using our proposed method. The horizontal axis represents the data points and the vertical axis represents the sensor magnitude.

correct value of the faulty sensor in the affected period). Both errors should be small. We found that the spillover error of our method is nearly constant around 0.52, while that of the prior work method increases from 0.60 (when the step magnitude = 20) to 7.1 (when the step magnitude = 100). Thus, our method is much better than prior work in terms of spillover reduction. The sensor estimation errors of both methods are good, comparable and do not vary much with the step magnitude (around 0.25 for our method and 0.45 for prior work method). We attribute the good performance of the prior work method to the fact that the 15 normal BPTC sensors used as inputs to estimate the faulty sensor are *all* normal sensors. It is expected that if more than one of these 16 BPTC sensors are faulty, the sensor estimation error for the prior work method will be large.

5. Conclusion

An advanced and reliable power plant monitoring system should be able to handle both out-of-range detection and fault detection. However, most prior work has ignored the out-of-range problem. We apply support vector representation machine to process driver sensors for out-of-range detection. In cases of data exceeding the model training range, the system notifies the operator to include more historical data which capture this new operating state to retrain the model to guarantee its accuracy. In fault detection, we propose to use hidden variables in regression models to reduce spillover. Test results show the advantage of the proposed method.

Besides power plants, the idea of using hidden variables can be extended for other regression applications, in which original system inputs are insufficient to estimate the outputs. In a more general scenario, where there is inadequate information about the system inputs and outputs, advanced correlation analysis is needed to determine the inputs and outputs for a regression model. Our future work will address this.

References

[1] J. W. Hines, A. Gribok and B. Rasmussen, "On-Line Sensor Calibration Verification: A Survey", the 14th International Congress and Exhibition on Condition Monitoring and Diagnostic Engineering Management, Manchester, England, September, 2001.

[2] D. Ruan and P. F. Fantoni, "Power Plant Surveillance and Diagnostics", Springer-Verlag, 2002.

[3] J. Ding, A. Gribok and J. W. Hines, "Redundant Sensor Calibration Monitoring Using ICA and PCA", Real Time Systems Special Issue on "Applications of Intelligent Real-Time Systems for Nuclear Engineering", accepted for publication.

[4] D. Tax and R. P. W. Duin, "Data Domain Description using Support Vectors", Proc. of European Symposium on Artificial Neural Networks, 1999, pp.251-256.

[5] C. Yuan and D. Casasent, "A Novel Support Vector Classifier with Better Rejection Performance", Int'l Conf. on Computer Vision and Pattern Recognition, 2003, pp.419-424.

[6] A. V. Gribok, A. M. Urmanov, J. W. Hines and R. E. Uhrig, "Use Of Kernel Based Techniques For Sensor Validation in Nuclear Power Plants", Statistical Data Mining and Knowledge Discovery, Chapman and Hall/CRC Press, 2004, pp. 221-235.

[7] V. N. Vapnik, "The Nature of Statistical Learning Theory", Springer-Verlag, New York, 1995.

[8] K. Chang and J. Ghosh, "A Unified Model for Probabilistic Principal Surfaces", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.23, No.1, 2001, pp.22-41.

[9] A. J. Smola and B. Scholkopf, "A Tutorial on Support Vector Regression", NeuroCOLT2 Technical Report Series, NC2-TR-1998-030, 1998.