Statistical Modeling of Fleet Data

Nicholas Moehle
Department of Mechanical Engineering, Stanford University
Advisor: Dimitry Gorinevsky

June 12, 2012

1 Introduction

This paper examines the probem of statistical modeling of fleet data. In other words, we hope to develop a tractable methodology for deriving a statistical model given a dataset that can be naturally divided into several similar groups (this notion will be made more rigorous later). This type of data might be natural for several intelligent energy problems of interest. We will consider a formulation that is extremely relevant to statistical monitoring, and can be applied to fleets of units (perhaps transformers, turbines, or other high-value equipment), each with its own data set. In this work, the performance modeling of a fleet of gas turbines is considered for the sake of example, but the reader is encouraged to consider other relevant applications.

1.1 Literature Review

Relatively little work has been in the area of fleet modeling. This paper follows the line of thought started in [1], [2], and [4]. The authors of [1] focus on linear regression, and specifically on choosing optimal regressors and formulating a real-time anomaly detection approach. In [2], these ideas are expanded to include drift anomalies and and explicit algorithm for distributed computation of the model parameters. In [4], the auther derives an update equation for the model parameters in real-time. Work in relevant areas includes [5], in which regression models are considered for *arrays* of data. Machine learning using a matrix normal distribution is discussed in [10]. Methods using PLS regression are related to the ideas presented below. A good summary of PLS is given in [8].

Statistical monitoring of gas is explored in [6] and [3], using a case-based reason method. Two standard approaches for anomaly detection in gas turbines are involve using a bank of Kalman filters or neural networks. A good reference for both is [9].

1.2 Dataset

The data set of interest is assumed to have the following hierarchy:

- *Unit* It is assumed that we have data describing N units, of similar design, from which we would like to learn about the performance of these units. The available data is divided into N subsets, with one subset for each unit.
- *Time* Each of the N subsets contains T data points. Although we will use the interpretation that the points are ordered and temporal, this assumption is not necessary. Furthormore, we assume that each unit has exactly T data points; this is done purely for convencience, and there is a natural and obvious generalization to cases in which the individual units each have a different number of data points (in this case, we replace the T in (1) with T_i).

- *Input/Output* Each data point is divided into input and output data, i.e. the system has an input output structure that is known *a priori*.
- *Multivariate data points* Each output data point $y_i(t)$ and input data point $x_i(t)$ is a vector in $\mathbb{R}^{n_y}$ and $\mathbb{R}^{n_x}$, respectively.

Therefore, we can summarize the data structure compactly by writing

$$\left\{\{x_i(t), y_i(t)\}_{t=1}^{T}\right\}_{i=1}^{N} \tag{1}$$

We will also use the notation $X_i \in \mathbb{R}^{n_x \times T}$ and $Y_i \in \mathbb{R}^{n_y \times T}$ to denote the data subsets $\{x_i(t)\}_{t=1}^{T}$ and $\{x_i(t)\}_{t=1}^{T}$, respectively, in matrix form.

1.3 Objectives

We propose a regression model of the following form:

$$y_i(t) = B_i x_i(t) + v_i(t) \tag{2}$$

with the following variables:

- $y_i(t) \in \mathbb{R}^{n_y}$ is the (known) output of unit i at time t.
- $x_i(t) \in \mathbb{R}^{n_x}$ is the (known) exogenous input for unit i at time t.

$B_i \in \mathbb{R}^{n \times k}$ is the (static, linear) model for the i^{th} unit. The term $v_i(t) \in \mathbb{R}^{n_y}$ represents noise and model mismatch, and we would somehow like to choose each B_i such that each $v_i(t)$ is small with respect to some meaningful norm. Furthermore, we would like to know about the covariance of v_i, for all i, which may be useful, for example, in classifying additional data as normal or anomalous.

Furthermore, we would like our approach to include the intuitive idea that the all the units are similar, i.e. B_i should be similar to B_j, for all $j \neq i$, and the covariance of v_i should be similar to the covariance of v_j, for all $j \neq i$, if, of course, these ideas are congruent with the actual data.

Finally, we desire that the approach be computationally feasible, meaning that the (globally) optimal solution can be computed with reasonable (and certainly polynomial) computational complexity.

2 Regression Model

We propose a regression model of the following form in (2) making further definitions for the following random variables:

- $v_i(t) \in \mathbb{R}^{n_y}$, where $v_i(t)|S \sim \mathcal{N}(0, S)$ is i.i.d., is the residual for unit i at time t. $S \in \mathbb{S}_+^{n_y}$ is the (unknown) covariance of the residual, for all units.
- $B_i \in \mathbb{R}^{n \times k}$, where $B_i|B_{\text{true}} \sim N_{n,k}(B_{\text{true}}, S/\alpha, I)$ is i.i.d., is the static linear map for unit i. The distribution $N(\cdot, \cdot, \cdot)$ denotes the matrix normal distribution, and in this case, simply means that each column of B_i is normally distributed, where the mean is the corresponding column of B_{true}, and the covariance is S/α. α is a tweak parameter , and is assumed to be known *a priori*.
- No prior information is given about B_{true} or S.

We seek to estimate the parameters B_{true}, S, B_i, for all $i = 1, ..., N$. In keeping with a typical regression problem formulation, we use the MAP method to estimate desired parameters. To do this, first we consider the density functions of the random variables of the problem, B_i, and $v_i(t)$:

$$f(v_i(t)|B_i, S) = \frac{1}{(2\pi)^{\frac{k}{2}}|S|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\Big(y_i(t) - B_i x_i(t)\Big)^T S^{-1}\Big(y_i(t) - B_i x_i(t)\Big)\right)$$
$$f(B_i|B_{\text{true}}) = \frac{1}{(2\pi)^{nk/2}|S|^{n/2}} \exp\left(-\frac{1}{2}\operatorname{tr}\left((B_i - B_{\text{true}})^T \alpha S^{-1}(B_i - B_{\text{true}})\right)\right) \tag{3}$$

The log likelihoods are, neglecting additive constants,

$$\log f(v_i(t)|B_i, S) = -(1/2)\log|S| - (1/2)\Big(y_i(t) - B_i x_i(t)\Big)^T S^{-1}\Big(y_i(t) - B_i x_i(t)\Big)$$
$$\log f(B_i|B_{\text{true}}) = -(n/2)\log|S| - \alpha(1/2)\operatorname{tr}\left((B_i - B_{\text{true}})^T S^{-1}(B_i - B_{\text{true}})\right)$$

We would like find the values of $B_1, ..., B_N$, B_{true}, and S which minimize the negative log likelihood function (this is equivalent to maximizing the likelihood function). Here we use $\mathcal{X}$ and $\mathcal{Y}$ to represent $X_1, ..., X_N$ and $Y_1, ..., Y_N$. Using the law of total probability,

$$\ell(B_1, ..., B_N, B_{\text{true}}, S|\mathcal{X}, \mathcal{Y}) = -\log f(\mathcal{X}, \mathcal{Y}|B_1, ..., B_N, B_{\text{true}}, S)$$
$$= -\log\left(\prod_{i=1}^{N} f(v_i(t)|B_1, ..., B_N, B_{\text{true}}, S)\right)$$
$$= -\log\left(\prod_{i=1}^{N} f(B_i|B_{\text{true}}, S)\prod_{t=1}^{T} f(v_i(t)|B_i, S)\right)$$

We have omitted the terms $f(B_{\text{true}})$ and $f(S)$, as there is no prior information about B_{true} or S. Distributing the logarithm:

$$= \sum_{i=1}^{N}\left(-\log f(B_i|B_{\text{true}}, S) - \sum_{t=1}^{T}\log f(v_i(t)|B_i, S)\right)$$

Plugging in the log-likelihoods from (3) and simplifying:

$$= NT\log|S| + \sum_{i=1}^{N}\left(\alpha\operatorname{tr}\left((B_i - B_{\text{true}})^T S^{-1}(B_i - B_{\text{true}})\right) + \operatorname{tr}\left((Y_i - B_i X_i)^T S^{-1}(Y_i - B_i X_i)\right)\right) \tag{4}$$

Although this function is not technically convex in its current form (in the variables B_i B_{true} and S), it *is* convex in the variables $S^{-1}B_i$, $S^{-1}B_{\text{true}}$, and S^{-1}, and a general solver could easily be used to find the solution. A simple analytical solution exists, however, which we derive below.

2.1 Normal Equations and Solutions

In this section, hope to analytically find the minimizers of (4). We will make heavy use of matrix calculus, and specifically, differentiation of scalar functions of matrices (functions $f : \mathbb{R} \to \mathbb{R}^{m\times n}$), where

$$\frac{\partial f(X)}{\partial X} = \begin{bmatrix} \frac{\partial f(X)}{\partial x_{11}} & \frac{\partial f(X)}{\partial x_{21}} & \dots & \frac{\partial f(X)}{\partial x_{m1}} \\ \frac{\partial f(X)}{\partial x_{12}} & \frac{\partial f(X)}{\partial x_{22}} & \dots & \frac{\partial f(X)}{\partial x_{m2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(X)}{\partial x_{1n}} & \frac{\partial f(X)}{\partial x_{2n}} & \dots & \frac{\partial f(X)}{\partial x_{mn}} \end{bmatrix}. \tag{5}$$

We perform partial minimization by using the familiar procedure of setting the partial derivative of a function equal to zero, which, in the matrix-valued case, amounts to setting all elements of (5) equal to zero. Performing this procedure for all variables in the problem is in fact equivalent to solving the KKT conditions of the problem (4).

2.1.1 Unit Model

Differentiating with respect to B_i,

$$\frac{\partial \ell}{\partial B_i} = 2\alpha S^{-1} B_i - 2\alpha S^{-1} B_{\text{true}} + 2S^{-1} B_i X_i X_i^T - 2S_i^{-1} Y_i X_i^T$$

Setting this equal to zero yields

$$\begin{aligned} Y_i X_i^T &= \alpha(\widehat{B}_i - B_{\text{true}}) + \widehat{B}_i X_i X_i^T \\ \widehat{B}_i &= (Y_i X_i^T + \alpha B_{\text{true}})(\alpha I + X_i X_i^T)^{-1} \end{aligned} \tag{6}$$

The above equation is very similar to the normal equations for linear regression, with the addition of the regularization term $B_i - B_{\text{true}}$. Note that (8) and (6) are both independent of S.

2.1.2 Average Model

Differentiating with respect to B_{true},

$$\frac{\partial \ell}{\partial B_{\text{true}}} = 2S^{-1} B_{\text{true}} - \sum_{i=1}^{N} 2S^{-1} B_i$$

Setting this equal to zero, the MLE estimate of B_{true} is

$$\widehat{B}_{\text{true}} = (1/N) \sum_{i=1}^{N} B_i \tag{7}$$

This has the intuitive interpretation of being the elementwise sample mean of all the individual unit models. Setting $B_i = \widehat{B}_i$, we obtain

$$\begin{aligned} \widehat{B}_{\text{true}} &= (1/N) \sum_{i=1}^{N} (Y_i X_i^T + \alpha B_{\text{true}})(\alpha I + X_i X_i^T)^{-1} \\ \widehat{B}_{\text{true}} &= (1/N) \left(\sum_{i=1}^{N} Y_i X_i^T (\alpha I + X_i X_i^T)^{-1} \right) \left(I - (\alpha/N) \sum_{i=1}^{N} (\alpha I + X_i X_i^T)^{-1} \right)^{-1} \end{aligned} \tag{8}$$

2.1.3 Residual Covariance

Finally we differentiate with respect to S:

$$\frac{\partial \ell}{\partial S} = NTS^{-1} + \sum_{i=1}^{N} \left(-\alpha S^{-1} (B_i - B_{\text{true}})(B_i - B_{\text{true}})^T S^{-1} - S^{-1} (Y_i - B_i X_i)(Y_i - B_i X_i)^T S^{-1} \right)$$

By setting equal to zero, we obtain

$$S = \frac{1}{NT} \sum_{i=1}^{N} \left(\alpha (B_i - B_{\text{true}})(B_i - B_{\text{true}})^T + (Y_i - B_i X_i)(Y_i - B_i X_i)^T \right) \tag{9}$$

2.2 Algorithm

The following is hardly even an algorithm, given its simplicity.

1. Compute $\widehat{B}_{\text{true}}$ from (8)
2. Compute $\widetilde{B}_i$ from (6)
3. Compute S from (9).

There are a couple of observations to make here. First, the value α adjusts how much coupling there is between the units, i.e. it sets the "strength" of the prior on B_i. This value can be adjusted to achieve desired performance.

Second, if we are only interested in computing $\widehat{B}_{\text{true}}$ and B_i, the algorithm is very amenable to a distributed setting. Consider the example of a distributed fleet of turbines, with a centralized monitoring center. To compute $\widehat{B}_{\text{true}}$, the centralized monitoring center needs only $Y_i X_i$ and $X_i X_i^T$, for all i, and not the entire dataset. This means that the monitoring center must be able to recieve $N(n_x^2+n_x n_y)$ double-precision numbers, and return $\widehat{B}_{\text{true}}$ (consisting of $N n_x n_y$ doubles) to the turbines in the fleet. This amounts to a very reasonable bandwidth. $\widetilde{B}_j$ can be computed locally. Step 3 would of course be much more difficult to carry out in this context.

3 Covariance Model

The above formulation may be suitable for some applications, but what if we are interested in knowing the residual covariances for each seperate unit? This is of concern for our anomaly detection problem; perhaps the turbines are much more different from each other than we originally expected (perhaps the turbines are operating in significantly diverse climates), and each has a radically different residual covariance structure. We would like to develop a formulation that is robust to these differences. Now consider the model (2) with the following additional definitions:

- $v_i(t) \in \mathbb{R}^{n_y}$, where $v_i(t)|S_i \sim \mathcal{N}(0, S_i)$ is i.i.d., is the residual for unit i at time t.
- $S_i \in \mathbb{S}_+^n$, where $S_i|S_{\text{true}} \sim \mathcal{W}(S_{\text{true}}/p, p)$ is i.i.d., is the covariance for unit i. The distribution $\mathcal{W}(\cdot,\cdot)$ is the Wishart distribution (if we take a random matrix $Z \in \mathbb{R}^{n\times\nu}$, where the columns of Z are normally distributed, zero-mean i.i.d. random vectors with covariance Σ, then $ZZ^T \sim \mathcal{W}(\Sigma,\nu)$; this is therefore a natural prior for a covariance matrix). $p \in \mathbb{R}$ is a tweak parameter, and is assumed to be known *a priori*.
- We have no prior information about B_i or S_{true}.

We seek to estimate the parameters S_{true}, and B_i and S_i, for all $i = 1, ..., N$. Again we use the maximum *a posteriori* probability (MAP) method to estimate desired parameters. To do this, first we consider the density functions of the random variables of the problem, S_i, and $v_i(t)$:

$$f(v_i(t)|S_i) = \frac{1}{(2\pi)^{\frac{k}{2}}|S_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(y_i(t)-B_i x_i(t))^T S_i^{-1}(y_i(t)-B_i x_i(t))\right)$$

$$f(S_i|S_{\text{true}}) = \frac{|S_i|^{\frac{p-n-1}{2}}}{2^{(np/2)}\,|pS_{\text{true}}|^{(p/2)}\,\Gamma_n(\frac{p}{2})} \exp\left(-\frac{1}{2}\text{tr}\left(pS_{\text{true}}^{-1}S_i\right)\right)$$

The log likelihoods are, again ignoring positive constants

$$\log f(v_i(t)|B_i,S) = -(1/2)\log|S| - (1/2)(y_i(t)-B_i x_i(t))^T S^{-1}(y_i(t)-B_i x_i(t))(t)$$

$$\log f(S_i|S_{\text{true}}) = (p/2)\log|S_{\text{true}}| - \frac{p-n-1}{2}\log|S_i| - \frac{p}{2}\text{tr}\left(S_{\text{true}}S_i^{-1}\right) \tag{10}$$

We would like to maximize the log likelihood function. Using the law of total probability,

$$\ell(S_1, ..., S_N, B_1, ..., B_N, S_{\text{true}}|\mathcal{X}, \mathcal{Y}) = -\log f(\mathcal{X}, \mathcal{Y}|S_1, ..., S_N, B_1, ..., B_N, S_{\text{true}})$$
$$= -\log\left(\prod_{i=1}^{N} f(v_i(t)|S_1, ..., S_N, B_1, ..., B_N, S_{\text{true}})\right)$$
$$= -\log\left(f(S_{\text{true}})\prod_{i=1}^{N} f(S_i|S_{\text{true}})f(B_i)\prod_{t=1}^{T} f(v_i(t)|S_i, B_i)\right)$$

As before, we ignore the terms $f(S_{\text{true}})$ and $f(B_i)$. Distributing the logarithm:

$$= \sum_{i=1}^{N}\left(-\log f(S_i|S_{\text{true}}) - \sum_{t=1}^{T}\log f(v_i(t)|S_i, B_i)\right)$$

Plugging in the log-likelihoods from equations (10) and simplifying:

$$= Np\log|S_{\text{true}}| + \sum_{i=1}^{N}\left(p\,\text{tr}\left(S_{\text{true}}^{-1}S_i\right) + (T+n+1-p)\log|S_i| + \text{tr}((Y_i - B_iX_i)^T S_i^{-1}(Y_i - B_iX_i))\right)$$

We need to make the change of variables, $S_i^{-1} = P_i$, $L^TL = S_{\text{true}}^{-1}$, and $\widetilde{B}_i = S_i^{-1}B_i$.

$$= -Np\log\left|L^TL\right| + \sum_{i=1}^{N}\Big(p\,\text{tr}\left(L^TP_i^{-1}L\right) - (T+n+1-p)\log|P_i|$$
$$+ \text{tr}(Y_i^TP_iY_i) - 2\,\text{tr}(Y_i^T\widetilde{B}_iX_i) + \text{tr}((\widetilde{B}_iX_i)^TP_i^{-1}(\widetilde{B}_iX_i))\Big) \tag{11}$$

The variables over which we optimize are P_i, L, and $\widetilde{B}_i$. This is convex for $1 + n + T > p$, which is reasonable, because in many problems of interest, T is very large.

3.0.1 Proof that (11) is convex

The function $\ell(L, P_i, \tilde{B}_i|\mathcal{X}, \mathcal{Y})$ is jointly convex in all its variables. It suffices to show that each term of the sum is convex in its variabls:

- Consider $-Np\log|L^TL| = -Np\log(|L^T|\cdot|L|) = -Np(\log|L| + \log|L|) = -2Np\log|L|$. The logarithm-determinant function is concave, so its additive inverse is convex.
- Consider $p\,\text{tr}(L^TP_i^{-1}L) = p\sum_{j=1}^{n}(Le_j)^TP_i^{-1}(Le_j)$, where the e_j are the unit vectors. This is the sum of matrix fractional functions, and is convex in both variables [Boyd pg. 76].
- Consider $-(1+n+T-p)\log|P_i|$. The logarithm-determinant function is concave, so for $1+n+T \geq p$, this is convex, and for $1 + n + T \leq p$, it is concave (for $1 + n + T = p$, it is zero).
- Consider $\text{tr}((\tilde{B}_iX_i)^TP_i^{-1}\tilde{B}_iX_i)$. Again, the matrix fractional function is convex.

3.1 Normal Equations and Solutions

Because (11) is convex, we could use a generalized solver to compute the solution. However, we can find a simple method to compute the global optimum, and gain some intuition into the problem, by using the solution outlined in this section. We use the same method as before: using partial differentiation to compute the normal equations, which can be solved interatively. This is analogous to a coordinate descent method, except in this case, the coordinates over which we optimize at each iteration is matrix valued.

3.1.1 Average Residual Covariance

Differentiating with respect to L, we have, from [7],

$$\frac{\partial \ell}{\partial L} = -2Np(L^{\dagger})^T + p\sum_{i=1}^{N} 2P_i^{-1}L$$

where † denotes the pseudoinverse. We assume L is invertible and set this equal to zero to obtain

$$(\widehat{L}^T\widehat{L})^{-1} = 1/N\sum_{i=1}^{N} P_i^{-1} \tag{12}$$

Making the change back to the natural variables yields $\widehat{S}_{\text{true}} = 1/N\sum_{i=1}^{N} S_i$.

3.1.2 Residual Covariance

Differentiating with respect to P_i,

$$\frac{\partial \ell}{\partial P_i} = -pP_i^{-1}LL^TP_i^{-1} - (1+n+T-p)P_i^{-1} + Y_iY_i^T - P_i^{-1}\tilde{B}_iX_iX_i^T\tilde{B}_i^TP^{-1}$$

Changing back into the natural variables and setting equal to zero gives:

$$0 = -p\widehat{S}_iS_{\text{true}}^{-1}\widehat{S}_i - (1+n+T-p)\widehat{S}_i + Y_iY_i^T - B_iX_iX_i^TB_i^T \tag{13}$$

If we consider another change of variables:

- $Q^{(r)} = Y_iY_i^T - B_iX_iX_i^TB_i^T$
- $A^{(r)} = -(1/2)(1+n+T-p)I$
- $B^{(r)} = I$
- $R^{(r)} = (1/p)S_{\text{true}}$

Then (13) can be solved as an algebraic Riccati equation (a well-known matrix quadratic equation with form $A^TX + XA - XBR^{-1}B^TX + Q = 0$), which can be solved cheaply.

3.1.3 Unit Model

Differentiating with respect to $\tilde{B}_i$,

$$\frac{\partial \ell}{\partial \tilde{B}_i} = 2P_i^{-1}\tilde{B}_iX_iX_i^T - 2Y_iX_i^T$$

Changing back to the natural coordinates and setting equal to zero gives

$$Y_iX_i^T = \widehat{B}_iX_iX_i^T \tag{14}$$

3.2 Algorithm

The following algorithm can be used to obtain S_i, S_{true}, and B_i for all i

1. Compute $\widehat{B}_i$ from (14)
2. Initialize $\widehat{S}_i$ and $\widehat{S}_{\text{true}}$.
3. Compute $\widehat{S}_i$ from (13)
4. Compute $\widehat{S}_{\text{true}}$ from (12).
5. Check the variables have converged. If so, stop. If not, go to step 3.

4 Simulation

In order to test these two formulations and briefly show their relative strengths,

Data were generated according to (2), with

- $x_i(t) \in \mathbb{R}^n$ $x_i(t) \sim \mathcal{N}(0, \Sigma_x)$, i.i.d. the (known) input of unit i at time t.
- $v_i(t) \in \mathbb{R}^n$, $v_i(t) \sim \mathcal{N}(0, S_i)$, i.i.d., is the residual for unit i at time t.
- $B_i \in \mathbb{R}^{n \times k}$, $B_i \sim N_{n,k}(B_{\text{true}}, S_i, I)$ is the static linear map for turbine i.
- $S_i \sim \mathcal{W}(S_{\text{true}}/p, p)$ is the covariance of the residual, with p degrees of freedom.

$$\Sigma_x = \begin{bmatrix} 1 & 0.5 & 0.25 & 0 \\ 0.5 & 1 & 0.5 & 0 \\ 0.25 & 0.25 & 1 & 0 \\ 0 & 0 & 0 & 0.001 \end{bmatrix} \qquad S_{\text{true}} = \begin{bmatrix} 0.1 & 0.01 \\ 0.01 & 0.001 \end{bmatrix}$$

$$B_{\text{true}} = \begin{bmatrix} -29.62 & -29.36 & 1 & 0.0733 \\ 0.0314 & 0.0385 & 0.947 & -9.51 \times 10^{-5} \end{bmatrix} \qquad T = 100 \qquad N = 50 \qquad p = 10 \qquad \alpha = 1$$

The data for B_{true} was obtained from [6] As a way of measuring the efficacy of the two different formulatioss, we plot $\|S_i - \widehat{S}_i\|$ the distance between estimated unit covariances, and the actual unit covariances, for all i, and we do similarly for $\|B_i - \widehat{B}_i\|$. The norm used in this case was the spectral norm (the Frobenius norm was avoided because it has a statistical interpretation). The results are shown in figures 1 and 2, and are compared with a naive approach, in which the all data is pooled together, and a single model and covariance is computed for the entire fleet. We see that $\|B_i - \widehat{B}_i\|$ is large for both the covariance model and the naive model, but for the regression model it is low. Also, $\|S_i - \widehat{S}_i\|$ is large for both the regression model and the naive model, but for the covariance model, it is low. This reflects the fact that the covariance model is the only model that takes individual unit covariances into account.

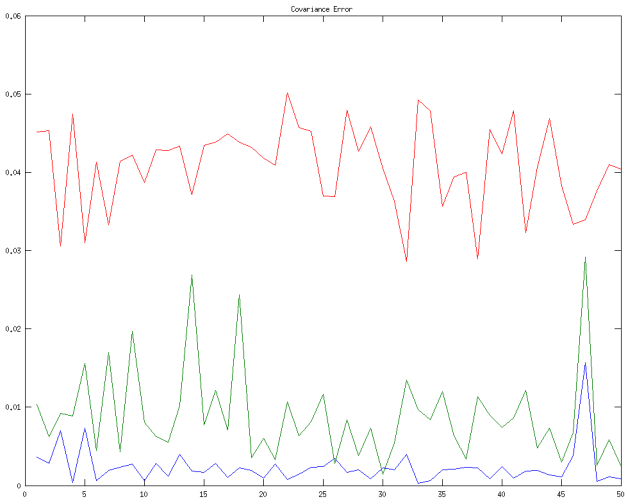

Figure 1: The error $\|S_i - \widehat{S}_i\|$ vs. i, for the regression model (green), the covariance model (blue), and the naive model (red)

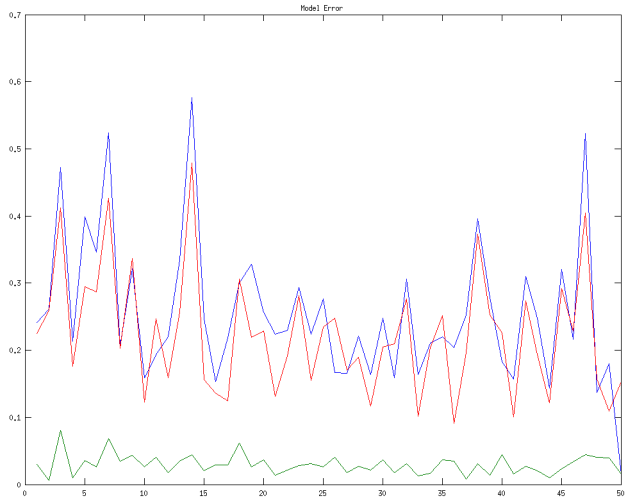

Figure 2: The error $\|B_i - \widehat{B}_i\|$ vs. i, for the regression model (green), the covariance model (blue), and the naive model (red)

4.1 Future Work

System identification represents the first, and possibly most important, step towards a feasible and practical anomaly detection methodology, and we believe the algorithms described above present sound approach to extract a system model from a dataset. However, the algorithms described in this paper, and the dataset that used to test them were, of course, both products of the same reasoning. A real test of the efficacy of these algorithms can only be done with real data, from, for example, a real fleet of gas turbines in operation. Because of the importance of using real data for validation, we are actively seeking sources of such data.

There is a second naive formulation, namely that of treating each unit individually, and ignoring the rest of the fleet data while making estimates of that unit's parameters. It remains to be shown that there are problems for which our two formulations are superior to this second naive formulation (although the author strongly believes this is true). Unfortunately, the brevity of the current academic term has prevented any such simulation from being realized.

The main question that remains to be answered is whether or not there exists a solution the the problem of estimating the unit models and the residual covariances simultaneously with prior parameters for both models and covariances. For the time being, it appears that the answer is "no," but this answer is not definitive.

5 References

References

[1] E. Chu, D. Gorinevsky, and S. Boyd. Detecting aircraft performance anomalies from cruise flight data. In *AIAA Infotech Aerospace Conference, Atlanta, GA*, 2010.

[2] E. Chu, D. Gorinevsky, and S.P. Boyd. Scalable statistical monitoring of fleet data. In *World Congress*, volume 18, pages 13227–13232, 2011.

[3] M. Devaney and B. Cheetham. Case-based reasoning for gas turbine diagnostics. In *AAAI 2005*, 2005.

[4] D. Gorinevsky. Bayesian fault isolation in multivariate statistical process monitoring dimitry gorinevsky. In *American Control Conference (ACC), 2011*, pages 1963–1968. IEEE, 2011.

[5] P. D. hoff. Separable covariance arrays via the Tucker product, with applications to multivariate relational data. *eprint arXiv:1008.2169*, August 2010.

[6] J. Petek and P. Hamilton. Performance monitoring of gas turbines. *Journal of Orbit*, 25(1):64–74, 2005.

[7] K.B. Petersen and M.S. Pedersen. The matrix cookbook. *Technical University of Denmark*, pages 7–15, 2008.

[8] R. Rosipal and N. Krämer. Overview and recent advances in partial least squares. *Subspace, Latent Structure and Feature Selection*, pages 34–51, 2006.

[9] AJ Volponi, H. DePold, R. Ganguli, and C. Daguang. The use of kalman filter and neural network methodologies in gas turbine performance diagnostics: a comparative study. *Journal of engineering for gas turbines and power*, 125:917, 2003.

[10] Y. Zhang and J. Schneider. Learning multiple tasks with a sparse matrix-normal penalty. *Advances in Neural Information Processing Systems*, 2010.