Support Vector Prognostics Analysis of Electronic Products and Systems

Vasilis A. Sotiris and Michael Pecht
Prognostics and Health Management Lab
Center for Advanced Life Cycle Engineering (CALCE)
University of Maryland, College Park 20742
vsotiris@calce.umd.edu

Abstract

This paper discusses the use of support vector machines (SVMs) to detect and predict the health of multivariate systems based on training data representative of healthy operating conditions. This paper also investigates a novel approach to SV classification and regression through the use of a principal component projection pursuit. Statistical indexes extracted from the reduced input space are used in a time series fashion for SV regression to predict the system health. The approach benefits from the reduced input space and from the small number of support vectors used to construct the classifier and predictor models, making it faster and robust. It is also immune to probabilistic assumptions and to the need for explicit models that describe the system behavior. A case study illustrates the use of support vector classification and regression. Case study results show that SVC correctly classified test points and minimized the number of false alarms and SVR correctly predicted function values for a predefined sinusoidal function. Together with excellent generalization ability, the proposed algorithm can be used in real time, making it a strong candidate for on-board, autonomous, system health monitoring, management and prediction.

Introduction

With increasing functional complexity of on-board autonomous systems, there is now an increasing demand for early system level health assessment, fault diagnostics, and prognostics. There are many techniques that have been explored to address this challenge, including environment and usage monitoring [1], binning and density estimation of Load Parameters [2], data-driven neural networks [17], as well as kalman filtering and Bayesian estimation techniques.

Health management and assessment of electronic systems is a complex task, largely because of the complexity of the electronic data, large number of parameters, competing failure mechanisms and presence of intermittent faults and failures. Traditional univariate analyses analyze each parameter separately whereas an SVM approach evaluates the data as a whole, trying to differentiate or characterize the mixture of information without necessarily requiring all the system parameters. This is especially true when working with complex electronic systems, where hundreds of signals can coexist in the sample and it is very difficult to identify every single contributor to the final system output. Each sensor can be sensitive to a range of environmental stimuli (surface insulation resistance (SIR), leakage current, cross talk EMI, currents and voltages, vibration frequency, temperature, electrical resistance, inductance) [4], [6] although with different sensitivities. At the same time, each single stimulus is sensed by more than one sensor since their sensitivities overlap. The SVM approach will take advantage of the overlapping sensitivities and process the information generated by these sensors to improve the resolution and accuracy of the analysis, be much more economical and easier to build.

In this paper we explored the use of support vector machines (SVMs) as a candidate for a new onboard autonomous data processing methodology. In particular for detecting intermittent faults in systems with multi-modal or unknown parameter distributions. This makes the use of SVMs very practical in providing system level health decisions with a minimal footprint on computational, memory and power resources. SVMs are also trained to optimize generalization with a reduced training set making the training of SVMs much simpler and economical.

Support vector machines (SVMs) are used for novelty detection through support vector classification (SVC) and prediction through support vector regression (SVR) in time series data collected from sensors. Figure 1 shows the flow diagram for the SVM algorithm. The input space (training data) is processed and reduced through projections using a principal component analysis, from which two lower dimensional subspaces are extracted. The model subspace is constructed through optimization and based on the parameters that exhibit the greatest variance in the input space. The residual subspace is chosen to be orthogonal to the model subspace and used to extract the inverse information with respect to the model space. SVMs are then trained using the distribution of the projected input data. From these subspaces statistical indexes T^2 and SPE are further extracted. T^2 and SPE are one-dimensional vectors which are fed into SVC and SVR respectively. SVC and SVR then detect and predict using the reduced

statistical indexes. Detection and prediction results are obtained independently from the model subspace path and from the residual path. The comparison of the results provides measures of accuracy and also information of false and positive alarms.

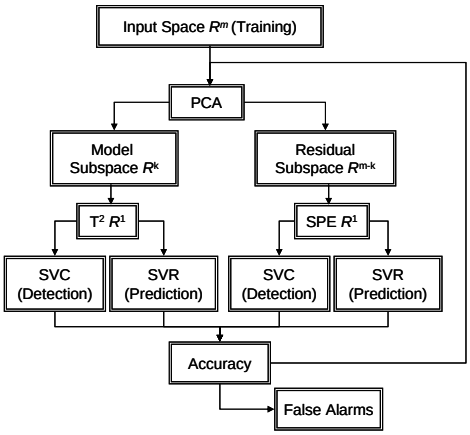

Figure 1: Flow chart diagram of support vector prognostic approach

Linear support vector classification theory

We first introduce the linear SVC theory starting with "hard margin classification" to build the framework for the "soft margin" and "nonlinear classification" in subsequent sections. In hard-margin classification, training data are linearly separable whereas in soft and nonlinear classification the training data are mostly not linearly separable. Suppose $\boldsymbol{x}$ is the input vector, y is the class label, d is the number of dimension and n is the number of samples. Training data (x_i,y_i) where $\boldsymbol{x} \in X^m$, $y_i \in \{+1,-1\}$ *and* $i=1,\ldots,n$ can be separated by the hyperplane decision function $D(x)$ with appropriate $\boldsymbol{w}$ and b:

$$D(x) = (w^T x) + b = \sum_{i=1}^{n} w_i x_i + b \qquad (1)$$

where $\boldsymbol{w}=[w_1,\ldots,w_n]^T$ is the weight vector of the hyperplane and $\boldsymbol{x}=[x_1,\ldots x_n]^T$. Thus, training data with $y_i=+1$ will fall into $D(x)>0$ while the others with $y_i=-1$ will fall into $D(x)<0$.

There are many possible separating hyperplanes which can classify the training samples, but we only need to choose one which is called the optimal separating hyperplane (OSH). Finding the (OSH) is important because it determines the accuracy of the detection and prediction process after the training. In detection the OSH defines the boundary in which new observations are considered normal/healthy and outside of which they are considered abnormal/unhealthy. In the prediction process the OSH is the model fit for the training data, and its extrapolation is used to predict future system health states. To determine the (OSH), support hyperplanes are used. The input vectors that pass through the support hyperplanes are called support vectors. The distance between two support hyperplanes is defined as the margin M. The separating hyperplane with the maximum margin is called the (OSH).

Support hyperplanes are parallel to the (OSH) $D(x)=w^Tx+b=0$, and their equations can be written as: $D(x)=w^Tx+b=k$ and $D(x)=w^Tx+b=-k$, where k is a positive integer. However, the above two equations are over-parameterized. If we multiply a constant to $\boldsymbol{w}$, b and k, the equations still hold. However, if we set $k=1$ so that only one set of $\boldsymbol{w}$ and b will be the solution the equations become: $D(x)=w^Tx+b=1$ and $D(x)=w^Tx+b=-1$. In hard-margin classification, no input data fall between two support hyperplanes. The training data are restricted to the following constraints: $w^Tx_i+b\geq+1$ for $y_i=+1$, and $w^Tx_i+b\leq-1$ for $y_i=-1$, $i=1,\ldots,n$, which can be rewritten as:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \geq 0 \quad \text{for} \quad i = 1,\ldots,n \qquad (2)$$

where n is the number of input vectors. The margin is then given by $M=2/||\boldsymbol{w}||$, called the objective function. Therefore, the maximal margin M can be found by minimizing $||\boldsymbol{w}||$. To simplify computations, the objective function is squared: $2/||\boldsymbol{w}||^2=1/2\boldsymbol{w}^T\boldsymbol{w}$ and minimized subject to the constraints $y_i(w^Tx_i+b)-1\geq0$ *for* $i=1,\ldots,n$. Lagrangian multipliers are used to convert the objective function from the primal space (input space) to a dual space to simplify the calculations. The primal form of the lagrangian function can be stated as:

$$L_P(\mathbf{w},b,\alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) + \sum_{i=1}^{n}\alpha_i \qquad (3)$$

The lagrangian functions in primal and dual spaces are written as L_P and L_D respectively. The idea is to minimize L_P with respect to $\mathbf{w}$ and b or to maximize L_D with respect to α. The optimal solution ($\boldsymbol{w}^*$, b^*, α^*) exists if and only if KKT conditions are satisfied. The KKT conditions state that: i) the partial derivative with respect to w and b is equal to zero ii) $\alpha_i[y_i(w^Tx_i+b)-1]=0$ and iii) $\alpha_i\geq0$ *for* $i=1,\ldots,n$. Taking the partial derivative with respect to $\boldsymbol{w}$ and b and setting the derivative to zero gives a set of equations to solve for $\boldsymbol{w}$ and b. By applying the KKT conditions we find w and b:

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i \qquad (4)$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0 \qquad (5)$$

which in turn can be used to formulate the dual optimization problem:

$$L_D(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i{}^T\mathbf{x}_j \qquad (6)$$

subject to (4). Quadratic Programming is used to solve for the optimal lagrange multipliers α^* in the dual form, through which $\boldsymbol{w}^*$ and b^* can be found. In matrix form the dual problem is expressed as

$$L_D(\alpha) = -\frac{1}{2}\alpha^T \mathbf{H}\alpha + p^T\alpha \quad (7)$$

where the Hessian matrix $\boldsymbol{H} = y_i y_j x_i^T x_j$, $\alpha=[\alpha_1,...,\alpha_n]$ and $p^T=[1,...,1]^T$ which has a size of (n x 1). The dual optimization formulation is: minimize (6) subject to (4) where $\alpha_i \geq 0$. To improve the estimate of the value of b, the average value of b is found by averaging over all support vectors:

$$b = \frac{1}{n_s}\sum_{i=1}^{n_s} y_i - \frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{j=1}^{n} \boldsymbol{\alpha}_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (8)$$

$$b = \frac{1}{n_s}\sum_{i=1}^{n_s} y_i - \frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{j=1}^{n} \mathbf{H}\boldsymbol{\alpha}_j \quad \text{where } \mathbf{H} = y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$b = \frac{1}{n_s}\sum_{i=1}^{n_s} y_i \left(1 - \sum_{j=1}^{n_s} \mathbf{H}_{ji}\boldsymbol{\alpha}_j\right) \quad (9)$$

In real world applications data are rarely linearly separable and therefore we are interested in the nonlinear classifier for use in both SVC and SVR. Before, we examined the theory for separable data (hard-margin), now we look at data that are inseparable (soft-margin). In this case an input data point can have an error ξ which is called the slack variable if it falls inside the margin. For $0< \xi_i<1$, the data are not well separated but still correctly classified and for $\xi_i>1$, data are misclassified. The summation of slack $\sum^{i=1,n}(\xi_i)$ is the upper bound on the errors. When the slack variable ξ is introduced, the constraints on (x_i, y_i) are always met, so feasible solutions always exist. Thus, we need to penalize the objective function by adding an error term to the optimization equation. The objective function $f(w,b)$ now becomes:

$$f(\mathbf{w}, b, \xi) = \frac{1}{2}\|\mathbf{w}\|^2 - C\sum_{i=1}^{n} \xi_i^p \quad (10)$$

subject to $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$ for $i = 1,...,n$ where C is the margin penalty parameter that determines the trade-off between the maximization of the margin and minimization of the classification error. The slack variables ξi together with the constraints ensure that the decision function $D(x)=w^Tx+b$ selected fits the training set: almost all the data points verify that $D(x)-b\geq 0$ (ie., $\xi i=0$), and are located inside R (Figure 2). Some data points however, are such that $\xi i>0$, these are the outliers. The number of outliers is kept low by minimizing $\sum^{i=1,n}(\xi_i)$. Moreover the term $1/2\|w\|^2$ ensures that $D(x)$ has a minimum norm, which results in minimum volume for R. The dual optimization problem is given by:

$$\min L_P(w, b, \alpha, \beta) = \min! \; {}^1\!/_2 \|w\|^2 + C\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \alpha_i \left[y_i\left(w^T x_i + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{n} \beta_i \xi_i \quad (11)$$

where α and β are the Lagrange multipliers for the original function $1/2\|w\|^2$ and the error term ξ respectively. The idea now is to minimize L_P with respect to w, b and ξ or maximize with respect to the non-negative Lagrange multiplier α and β. The Karush-Kuhn-Tucker (KKT) conditions give (4), (5) and

$$\alpha_i + \beta_i = C \quad \text{for} \quad i = 1,...,n \quad (12)$$

The dual formulation is given by:

$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (13)$$

subject to (5) and $C \leq \alpha_i \leq 0$ for $i = 1,...,n$

In order to find the optimal hyperplane $D(x)$, a dual Lagrangian $L_D(\alpha)$ has to be maximized with respect to the non-negative Lagrange multiplier α by Quadratic Programming. When α_i=0, β_i=C which is a positive number, $\xi_i=0$, which means that the input data $\mathbf{x}_i$ is correctly classified and it is under the constraint $y_i(w^Tx_i + b)-1\geq 0$. When α_i=C, $y_i(w^Tx_i + b)-1+\xi_i=0$ and $\xi_i\geq 0$. The input vector x_i corresponding to $\alpha_i=C$ is called a bounded support vector which might be inseparable or misclassified, that is $y_i(w^Tx_i + b)-1\geq 0$. If $0<\xi_i<1$, x_i is correctly classified. If $\xi_i\geq 0$, x_i is misclassified. When $C<\alpha_i<0$, $y_i(w^Tx_i+b)-1+\xi_i=0$ and $\xi_i=0$. Thus, $y_i(w^Tx_i + b)=1$ and x_i is the unbounded support vector.

The nonlinear classifier

The basic idea in designing nonlinear SV machines is to map input vector $\boldsymbol{x} \in X^m$ into vectors $\Phi(x)$ of a higher dimensional *feature space F* (where Φ represents mapping $X^m \rightarrow X^f$), and to solve a linear classification problem as developed in the preceding theory in this feature space. The nonlinear decision function $D(x)$ is given by

$$D(x) = sign\left(\sum_{i=1}^{n} y_i \alpha_i \Phi^T(x_i)\Phi(x) + b\right) = sign\left(\sum_{i=1}^{n} y_i \alpha_i k(x_i, x) + b\right) \quad (14)$$

where k is the kernel function and *sign* decides the membership of the data point between the two classes. The mathematical formulation for the nonlinear classifier is solved through the same formulation as with the linear classifier. The difference is that the input space dot product is replaced by a new dot product defined by a chosen kernel function k. With the use of a kernel trick the mapping to the higher dimension can be accomplished though a dot product manipulation of the input space. The very efficiency of the SVC comes from Vapnik's principle: instead of designing $D(x)$ from an estimated underlying density, we design $D(x)$ directly. This avoids devoting unnecessary estimation accuracy to regions located far away from the decision boundary of $D(x)$ (The limiting hypersurface in X^m enclosing R) and to devote high estimation accuracy to regions close to the boundary.

Novelty detection through SVC

Novelty detection using SVC addresses the following problem: given a set of vectors $\boldsymbol{x}=[x_1,\ldots,x_m]^T$ in X^m such that $[x_1,\ldots,x_m]^T \sim d_0$, with d_0 unknown, is a new vector $\mathbf{x} \in X^m$ distributed according to d_0, and is considered normal under hypothesis H_0, and abnormal or "novel" under hypothesis H_1. In SVC, this problem is addressed through designing a decision function $D(x)$ (shown as the solid line in Figure 2) over region R in X^m and a real number b such that $D(x)-b \geq 0$, if $\boldsymbol{x} \in R$ and $D(x)-b<0$ otherwise. From the illustration in Figure 2 the points $\boldsymbol{x}$ which fall outside of the decision boundary are taken as outliers or abnormal observations. Notice the better classification performance obtained using the SV decision function. The decision function $D(x)$ is designed under two constraints: firstly, most of the training vectors $x=[x_1,\ldots,x_m]^T$ should be in R, except for a small fraction of abnormal vectors, called outliers, and secondly, it must be such that R in X^m has minimum volume. In order to estimate R, or equivalently $D(x)$ and b, we use a kernel function k in a higher dimensional space F. Space F can be implicitly selected by first choosing a positive definite kernel function k. A common choice is the Gaussian RBF kernel (where $\| \cdot \|_{\Phi}$ denotes the norm in X^m)

$$k(x_1, x_2) = e^{-\frac{\|x_1 - x_2\|^2}{2\sigma^2}} \tag{15}$$

A positive definite kernel k induces a linear feature space F of functions that utilize a dot product. Given a positive definite kernel k and the corresponding feature space F, the support vector novelty detection approach finds a linear optimal separating hyperplane in F that can be mapped back to the input space X^m as a nonlinear function resulting in R.

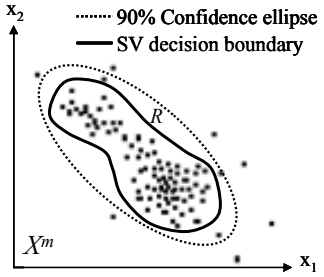

Figure 2: Illustration of the SV and 90% confidence decision boundaries

Support vector classification for novelty detection is a 2-steps process: 1) model construction: the machine is trained by a group of training data to construct a model from the 'experience', 2) model usage, which involves using the model constructed to classify the unknown data. In step 2, space R defines the classification model, where points that lie outside of the space ($D(x)<0$) defined by R are considered abnormal and points that fall within ($D(x)>0$) are considered normal. The natural log of (16) can also be used as an index, in which case an abnormal event should be detected whenever the index is greater than zero, which corresponds to $D(x)<0$. In comparison to the Gaussian 90% confidence boundary the SV boundary has the advantage that it is also dynamic, changing the decision boundary based on the changing shape of the data distribution, irrespective of the distribution of the data.

Prediction through SVR

Support vector regression (SVR) is used for prediction by extrapolating an optimal regression hyperplane (ORH) that best fits the training data. The extrapolation of the ORH is used to make predictions about the time dependent component of the data to estimate the health and remaining useful life for the system. SVR utilizes the mathematical framework for SVC to find the ORH. In SVR, we estimate the functional dependence of the dependent (output) variable $y \in X^m$ on an m-dimensional input variable $\boldsymbol{x}$. Therefore, unlike in SVC where the desired outputs y_i are discrete values e.g., Boolean, we deal with real valued functions. The learning stage ends in the same shape of a dual Lagrangian as in classification, only difference being in the dimensionalities of the Hessian matrix H and corresponding vectors, $\boldsymbol{H}=[G,-G;-G,G]$, where G is the grammian matrix and each element of this matrix is a corresponding kernel such that $G_{ij}=\Phi^T(x_i)\Phi(x_j)=k(x_i,x_j)$, $i,j=1,\ldots,n$.

The mathematical formulation for SVR is relatively unchanged as compared to that for SVC. In SVR we minimize the objective function called R given by:

$$R = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} |y_i - D(x_i, w)|_{\varepsilon} \tag{16}$$

where y are the measured values, C is a constant that influences a trade-off between an approximation error and the weight vector norm $\|w\|$, which is chosen by the user, and $D(x_i,w)$ is the model function value based on the ORH. The difference between y_i and $D(x_i,w)$ defines the error of the model at each observation i. An increase in C penalizes larger errors. Another design parameter chosen by the user is the required precision reflected in an ε value that defines the size of an error tube (ε-tube). Training data outside the tube have an associated slack error $\xi_i=|y-D(x,w)|-\varepsilon$, and training data inside have no error $\xi_i=0$. When ε is chosen to be large, the optimization process is allowed to fit a flatter prediction line, whereas when it is chosen small the fit is forced to be more nonlinear (less flat). It is important to optimize between a large ε-tube (greater uncertainty in prediction) and the linearity of the predictor function (less accurate prediction). This idea is illustrated in Figure 3; the prediction of an SVR is compared to the prediction of a linear regression. Also, the ε-tube associated with that prediction will also provide the uncertainty bounds on the prediction.

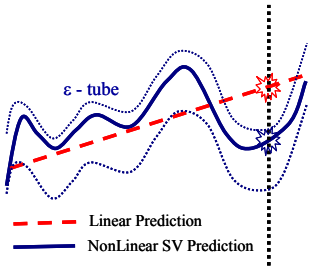

Figure 3: Comparison of an SVR prediction to a linear regression prediction

The minimization of the objective function R equals the minimization of the following:

$$R_{w,\xi} = \frac{1}{2}\|w\|^2 + C\left(\sum_{i=1}^{n}\xi_i\right) \qquad (17)$$

under the constraints: y_i-$w^T x_i$-$b \le \varepsilon + \xi_i$, $i=1\ldots n$ and $\xi_i \ge 0$. Similar to the procedures applied in the SVC, we solve the constrained optimization problem above by forming a primal variables lagrangian. The primal variables Lagrangian has to be minimized with respect to w,b and ξ_i and maximized with respect to nonnegative lagrange multipliers α and β.

Input space reduction and pattern recognition

In many electronic systems there are a large number of parameters (many of which are dependent) that can be analyzed to assess the system health and make prognosis. However, for in-situ (on-board) diagnostics and prognostics, it is generally necessary to compress the data from the parameters to extract the useful information. We do this by using a data compression technique based of the principal component analysis. This method is used to summarize the information into smaller representative dimensions.

A model is constructed through a principal component decomposition of the input space correlation matrix P. The model is designed to capture the variance related behavior of the system parameters, to considerably reduce the dimension of the input space and to discriminate highly correlated parameters by placing less importance on them. The model will be complemented by an orthogonal model subspace, the residual subspace. The residual subspace will take advantage of the inherently available inverse information. The new input space for the training stage for SVC and SVR will be the statistical indexes obtained from the projected data onto the model and residual subspaces respectively. Figure 4 illustrates the principal components, the projection onto the model and residual subspaces [S] and [R] respectively and the statistical indexes. Before a new observation is analyzed by SVC and SVR, it is projected onto these subspaces and its projection coordinates are used instead. The indexes are called the Hotelling T^2 and squared prediction error (SPE) as illustrated in Figure 4 (right). These indexes capture the multivariate and correlated information in each subspace and are used as a health index for the system. SVR can use the resulting one –dimensional index vectors to predict the health state of the system as a whole.

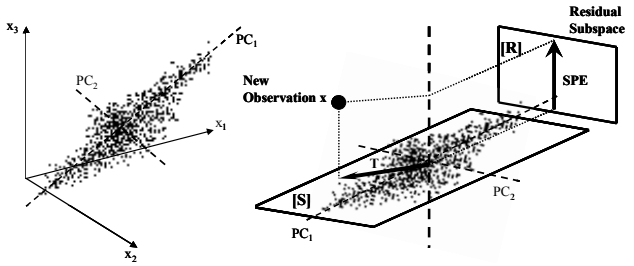

Figure 4: Geometric interpretation of: principal components, hotelling T^2 and SPE in the model [S] and residual [R] subspaces. T^2 and SPE are in turn used by SVC and SVR respectively.

The subspace decomposition into principal components can be accomplished using singular value decomposition (SVD) of the input space matrix $\boldsymbol{X}$ [16]. The SVD of data matrix $\boldsymbol{X}$, is expressed as $\boldsymbol{X}=USV^T$, where $S=diag(s_1,\ldots,s_m) \in R^{nxm}$, and $s_1>s_2>\ldots>s_m$. The two orthogonal matrices U and V are called the left and right eigen matrices of $\boldsymbol{X}$. Based on the singular value decomposition, the subspace decomposition of $\boldsymbol{X}$ is expressed as:

$$X = X_s + X_r = U_s S_s V_s^T + U_r S_r V_r^T \qquad (18)$$

The diagonal matrix S_s are the singular values $\{s_1,\ldots,s_k\}$, and $\{s_{k+1},\ldots,s_m\}$ belong to the diagonals of S_r. The set of orthonormal vectors $U_s=[u_1,u_2,\ldots,u_k]$ form the bases of signal space S_s. The original data is decomposed into three matrices, U, S and V, where matrix U contains the transpose of the covariance eigenvectors. The original data $\boldsymbol{X}$ is projected onto the signal subspace as defined by the PCA model. In terms of the SVD of $\boldsymbol{X}$, the projection matrix H can be expressed as UU^T where $u=USV^T$. Then the projection of vector $\boldsymbol{x}$ onto the model subspace can be expressed as $\boldsymbol{x}_{[S]}=UU^T\boldsymbol{x}$ and onto the residual subspace as $\boldsymbol{x}_{[R]}=(I-UU^T)\boldsymbol{x}$. Any vector $\boldsymbol{x}$ can then be represented by a summation of two projection vectors from subspaces S_s and S_r.

$$x = x_{[s]} + x_{[R]} = P_s\vec{x} + (I - P_R)\vec{x} \qquad (19)$$

where $P_S=UU^T$ and $P_R=I-UU^T$ are the projection matrices for the model and residual subspace respectively. In this framework, we can apply SVC and SVR having oriented the data such that we can better capture system faults (including intermittent) that are due to changes in variance, changes in correlation and changes in the distribution of the data. In this framework we can take advantage of the residual subspace and apply SVC and SVR to the residual subspace training data.

For a new sample vector $\mathbf{x}$, the Hotelling T^2 index is expressed as

$$T^2 = \vec{x}^T U_s P^{-1} U_s{}^T \vec{x} \quad (20)$$

where P is the covariance of $\boldsymbol{X}$, and is equal to $U^T U$.

The second statistic, the squared prediction error (*SPE*), indicates how well each sample conforms to the PCA model, measured by the projection of the sample vector onto the residual subspace

$$SPE = \left\| P_r \vec{x} \right\|^2 = r = \left\| (I - P_s)\vec{x} \right\|^2 \quad (21)$$

The new observation $\boldsymbol{x}$ is considered normal if SPE$\leq\delta^2$ and $T^2 \leq \tau^2$ where δ^2 and τ^2 are the control limits for the *SPE* and T^2 statistics, respectively, given a (1-α)confidence level [15]. These limits assume that **x** follows a normal distribution and T^2 follows a χ^2 distribution with k degrees of freedom, where k is defined to be the cut off for the number of principal components used in the PCA model. The control limits δ^2 and τ^2 are estimated based on the assumption that the data adhere to a Gaussian distribution which can lead to gross errors in processes that are highly non-linear. SVC is ideal for estimating the control limit using the same data and without any assumptions about their distribution.

SVC performance validation

To validate the performance of the SVC, simulated data are generated for two parameters x_1 and x_2, which are sampled dependently from a multivariate correlated distribution with a correlation matrix $\Sigma=[1,1.5;1.5,3]$, and a standard deviation σ and mean μ: $\sigma=1$, $\mu=[0,0]^T$. Outliers are generated artificially with a uniform random distribution that surrounds the "normal" training data within +/-3.5σ. Figure 5 shows the distribution of the training input space, where the normal training data are represented by the triangles and the abnormal training data by the squares.

A Gaussian RBF kernel function is used to generate the decision boundary *D(x)*, and the classification of an individual data point $\boldsymbol{x}$ is done by evaluating the sign of *D(x)* at that $\boldsymbol{x}$. Outliers or abnormal observations $\boldsymbol{x}$ are those who have negative values of *D(x)*. The dashed line on either side of the decision boundary (solid line) indicates regions in which new observations that fall in this region have membership with associated probabilities. New observations that fall distinctly outside of these dashed lines have a 100% membership into each category (healthy or not healthy).

Figure 6, shows the results of the SVC using the simulated data. The crosses correspond to normal training data and the circles the abnormal training data. Using the Gaussian RBF kernel we were able to define islands that represent the "normal" system state. A test vector $\boldsymbol{t}=[t_1,t_2,t_3,t_4]^T=[0\ 0;-2.5\ -3.1;2.1\ 3.5;-6\ -4]$ is used to test *D(x)*, where $\boldsymbol{t}$ is chosen such that the first point is a normal point and the other three are abnormal. The results show that the SVC algorithm detected the abnormal data in the test vector; points 2-4 and the one normal point.

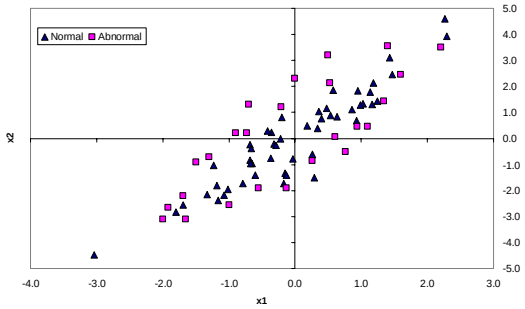

Figure 5: Input space for training data; normal (triangles) and abnormal (squares)

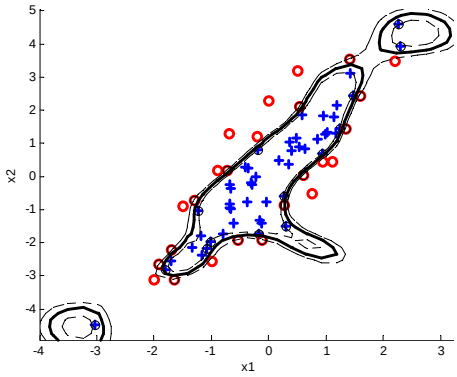

Figure 6: SVC decision function *D(x)* (solid line) and support vector boundary (dashed lines), RBF kernel, *C=500*

False and positive alarms

The user-defined parameters in SVC are the penalty parameter C, the kernel function k, and the degree of the function p. The decision boundary function *D(x)* is estimated differently depending on the value chosen for these parameters. The ideal decision function is one that minimizes the number of false and positive alarms. The penalty parameter C was varied experimentally to test for false positives, and false alarms. A true alarm is decided if an artificial outlier is detected. A false alarm is decided whenever a normal training data point is detected as an outlier/abnormal. Figure 7 shows the results for the false positive tests obtained by varying C from 0 to 500. After a value of C=175, the number of misclassified abnormal training points remains constant at 1. The misclassified point is found at: *[0.2600,-0.8700]*. Depending on the data structure and the correlation between parameters the penalty parameter will need to be empirically selected such that it minimizes the number of false alarms and false positives in the given training data set. A kernel function k is selected based on a comparison between three kernel functions: a linear, a polynomial of degree 2, and the Gaussian RBF. The number of false alarms and false positives were computed for each selection of kernel

function and found that the Gaussian RBF kernel performed the best.

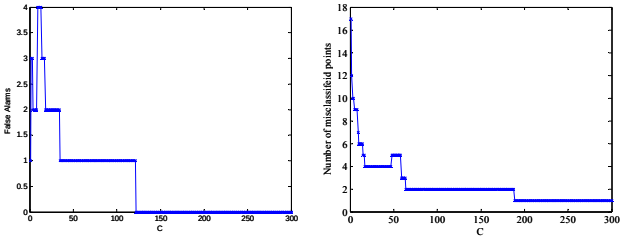

Figure 7: Number of false alarms (left) and number of misclassified abnormal training points (right) VS C

SVR performance validation

The performance validation of the SVR was studied by modeling a known sinusoidal function $f(x)=x^2 sin(x)$ with an added Gaussian noise, corrupted by 25% of the function standard deviation, with a zero mean. The model called the function estimator is used to predict future values of $f(x)$. The three most relevant parameters for SVR are also examined: the ε-insensitivity parameter, the penalty parameter (as in classification) and the shape parameters of the kernel function (variance, order of the polynomial). Again all three of these parameters are set by the user. Through experimentation it was found that for not too noisy data, the penalty parameter C can for all practical purposes be set to infinity (a very large number) and the regression estimation can be controlled by changing the insensitivity zone ε and shape parameters only. Figure 8 and Figure 9 show the implementation of the SVR and the effect of changing the ε-insensitivity parameter while keeping C very large.

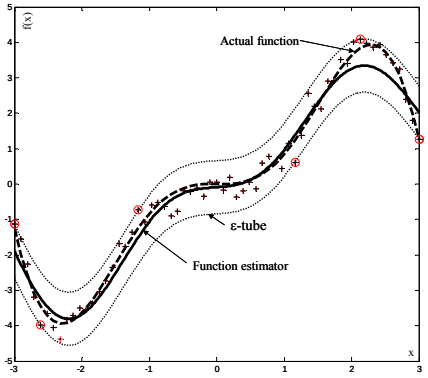

Figure 8: The influence of an insensitivity zone ε=0.75 on the regression model performance

The solid line shows the SVR estimate of the function using a radial basis function kernel (RBF). Figure 9 shows the estimator function when $\varepsilon=1$ and resulted in 13 support vectors whereas the plot in Figure 8 shows the estimator function when $\varepsilon=0.75$ and resulted in 33 support vectors. When ε is smaller the 33 chosen support vectors produce a better approximation to the true function given the noisiness of the data, although at the expense of larger support vector data set.

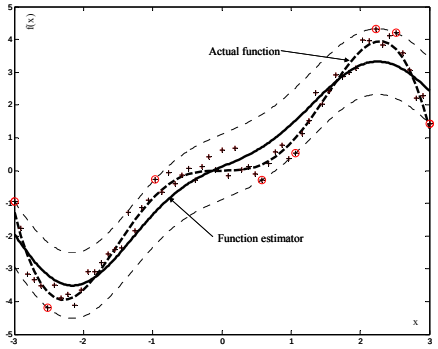

Figure 9: The influence of an insensitivity zone ε=1 on the regression model performance

When ε is chosen larger the estimation is not as good but the number of necessary support vectors is reduced. The uncertainty involved in both estimations is also different. The approximation in the Figure 9 is accompanied with a larger uncertainty (illustrated and quantified by the area/volume contained in the ε-tube), whereas with a smaller insensitivity zone the uncertainty is reduced (Figure 8).

The function estimator is then used to predict the value of the function at a future $\boldsymbol{x}$ (see Figure 10). The dashed line shows the actual function value of $f(x)$ and the solid black line shows the predicted values given the training data (+ signs in the figure). The prediction is based on the estimator model developed using training data up to $x=15$ (indicated by the dashed vertical line). The predictions for this example used $\varepsilon=0.1$ and $C=10e+5$. We found that further out predictions require more support vectors and suffer from higher uncertainty and lower accuracy. Better predictions are made when the time window is kept short.

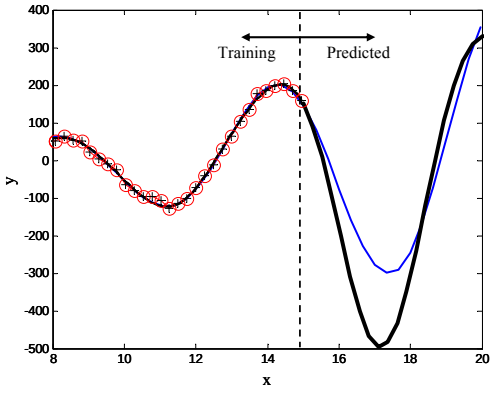

Figure 10: SVR predicted function values for x=20 given training information of x=0 to x=15.

Summary and conclusions

Support vector prognostics can be used to monitor, detect and predict system level faults and failures. This

paper presents a new approach to prognostics using SVM and SVR with the statistical indexes derived from a principal component projection pursuit of the input space.

In this approach, variance related features from the data are extracted and the classification through SVC is made more sensitive towards abrupt changes in variance as a means to detect intermittent faults. SVC is used to estimate the control limits for the reduced input space and detect abnormalities and SVR is used to analyze the statistical indexes to estimate the remaining useful life.

The implications of this approach are that: a) The control limits or otherwise the decision boundary for the distribution of the normal/healthy statistical indexes are computed based on the data and not based on assumptions regarding the distribution of the data. This makes both detection and prediction more accurate and adaptable to changes in the data. b) The nonlinear decision boundary estimated by SVC improves the detection performance and minimizes false alarms, especially in nonlinear data distributions. c) The summarization of the system health into a univariate statistical index (T^2 and SPE) is advantageously used by the SVR to estimate future values of the index and in effect the health of the system. Also this summarization is advantageous for faster autonomous online data processing. d) SVR analyzes all parameters simultaneously and gains considerable computational efficiency, and e) uncertainty calculations put a bound on the SVR prediction giving a probability distribution for each prediction.

In this paper we investigate the use of both SVC and SVR and discuss the application of SVC and SVR in combination with a PC projection technique. Simulated data were used for a case study and results were obtained for SVC and SVR respectively. Model parameter sensitivities were examined and tested experimentally and the results of the analysis showed that SVC was able to find the decision boundary for detection while SVR was able to predict a given noisy function value.

References

[1] Vichare, N., Rodger, P., Eveloy, N., and Pecht, M. 2006. Environment and Usage Monitoring of Electronic Products for Health Assessment and Product Design. *International Journal of Quality Technology and Quantitative Management.*

[2] Vichare, N., Rodgers, P., and Pecht, M. 2006. Methods for Binning and Density Estimation of Load Parameters for Prognostics and Health Management. *International Journal of Performability Engineering* 2(2).

[3] Pecht, M., Rogers, K., and Hillman, C. 2001. Hollow Fibers can Accelerate Conductive Filament Formation. *ASM International Practical Failure Analysis* 4(1), 57-60.

[4] Zhan, S., Azarian, M., and Pecht, M. 2006. Surface Insulation Resistance of Conformally Coated Printed Circuit Boards Processed With No-Clean Flux. IEEE Transactions on Electronics Packaging Manufacturing, 3(29), 217-233.

[5] Ganguly, G., and Dasgupta, A. 1994. Modeling PTH Damage Caused by Insertion Mount Connectors. *International Mechanical Engineering Congress & Exhibition*.

[6] Deng, Y., Pecht, M., and Rogers, K. 2006. Analysis of Phosphorus Flame Retardant Induced Leakage Currents in IC Packages Using SQUID Microscopy. *IEEE Transactions on Compaenents and Packaging Technologies*, 4(29), 804-808.

[7] Schoen, R.R., Lin, B.K., Habetler, T.K., Schlag, J.H., and Faraq, S. 1995. An unsupervised, on-line system for induction motor fault detection using Stator current monitoring. *IEEE Transactions Industry Appl*.31(6), 1274-1279.

[8] Tallam, R. M., Habetler, T. G., and Harley, R. G. 2002. Self-commissioning training algorithms for neural networks with applications to electric machine fault diagnostics. *IEEE transactions Power Electronics* 17(6).

[9] Mangina, E. E., McArthur, S. D. J., McDonald, J. R., and Moyes, A. 2001. A multi-agent system for monitoring industrial gas turbine start-up sequences. IEEE Transactions On Power Systems 16(3), 396-401.

[10] Godsill, S. J., and Rayner, P. J. W. 1998. Digital Audio restoration-A Statistical Model-Based Approach. London: Springer

[11] Davy, M., Desobry, F., Gretton, A., and Doncarli, C. 2005. An online support vector machine for abnormal events detection. *Signal Processing*, Elsevier.

[12] Smola, A., and Schoelkopf, B. 2002. *Learning with Kernels*. Cambridge, MA, USA: MIT Press

[13] Hayton, P., Schoelkopf, B, Tarasenko, L., and Anuzis, P. 2000. Support vector novelty detection applied to jet engine vibration spectra. *Neural Information Processing Systems*, 946-952

[14] Schoelkopf, B., Platt, J., Shaw-Taylor, J., Smola, A., and Williamson, R. C. 1999. Estimating the support in a high-dimensional distribution. *Technical Report TR87, Microsoft Research*.

[15] Edward, J., and Mudholkar, G. S. 1979. Control Procedures for Residuals Associated With Principal Component Analysis. *Technometrics* 3(21), 341-349.

[16] Haifeng, C., Guofei, J., Ungureanu, C., and Yoshihira, K. 2005. Failure Detection and Localization in Component Based Systems by Online Tracking. *KDD*, 750-755

[17] Carl, S., Byington, P.E., Watson, W., and Edwards, D. 2004. Data-Driven Neural Network Methodology to Remaining Life Predictions for Aircraft Actuator Components. *IEEE Aerospace Conference Proceedings*, 3581-3589