

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

185,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Fault Detection Method based on Extended Locally Linear Embedding

Wei Zhang and Xu Yang

*Shenyang Institute of Automation (SIA), Chinese Academy of Science (CAS)  
China*

## 1. Introduction

To enhance the autonomous capability, a system must be equipped with adequate numbers of the sensors responsible for the detection of system's input/output states, operating status, environment conditions, etc. However, more sensors may introduce other problems – the certainty, reliability, and robustness. Therefore, the reliable detection of sensor faults becomes extremely critical in order to assure system certainty, reliability, stability, and robustness. In general, the raw data obtained from the sensors of the complex systems have high dimensionalities and severe nonlinearity. High-dimensional raw data typically bear considerable redundancies and correlations hiding important relationships. It is, therefore, highly desirable to obtain more descriptive, informative, and useful data representations for the subsequent operations by eliminating the redundancies from the raw data. This process is referred to as dimensionality reduction. Dimensionality reduction can be achieved by creating a set of low-dimensional data representation with new features based on the transformations and/or combinations of the original data features. This operation is often referred to as “feature extraction”.

By far, the most frequently-used feature extraction method is Principle Component Analysis (PCA), which was proposed by Pearson in 1901<sup>[1]</sup> when he was studying regression analysis, and was mathematically developed by Hotelling in 1933<sup>[2]</sup>. PCA has been widely used in pattern recognition and machine learning. However, early dimensionality reduction methods were developed to primarily target the linear data systems <sup>[3][4]</sup>. When data system is high-dimensional and severely non-linear, these methods become ineffective.

In 1984 Hastie proposed concept of the principle curves<sup>[5]</sup>, which extended the PCA analysis to nonlinear fields. The principle curves can more accurately represent the nonlinear structures of data in linear dimensions. In 2000, Roweis proposed a manifold learning algorithm called Local Linear Embedding (LLE), which is an unsupervised non-linear technique that analyzes the high-dimensional data sets and reduces their dimensionalities with preserved local topology<sup>[6]</sup>. Today, LLE has been widely used in cluster analysis, image processing, biological informatics, etc.

## 2. Locally Linear Embedding Algorithm

The basic concept of LLE is to find a weight vector between a sample and its neighbors, and to keep this relationship in a feature space[6]. It assumes that even if the manifold embedded in a high-dimensional space is nonlinear, it still can be considered locally linear if each data point and its neighbors lie on or close to a locally linear patch of the manifold, i.e., the manifold can be covered with a set of locally linear patches which, when analyzed together, can yield information about the global geometry of the manifold. The weight vector expressing the intrinsic geometrical properties of the local patch can be obtained in three steps: (a) to find the neighbors of every sample in the high-dimensional space, (b) to obtain the reconstruction weight and a sparse matrix of the weight vectors, and (c) to compute the low dimensional embedding -- the bottom nonzero eigenvectors of the sparse matrix are the low dimensional embeddings of high dimensional samples.

Firstly, given a dataset  $X = \{x_1, \dots, x_n\}$ ,  $x \in R^d$ , assuming the data lies on a nonlinear manifold which locally can be approximated linearly, the cost function can be written as:

$$\min \varepsilon(W) = \sum_{i=1}^n \left| x_i - \sum_{j=1}^k w_j^i x_{ij} \right|^2 \quad (1)$$

Where  $x_{ij}$  is the neighbor of  $x_i$ , and  $w_j^i$  is the weight which subject to the constraint of  $\sum_{j=1}^k w_j^i = 1$ . It can be seen that the cost function means how well each  $x_i$  can be linearly reconstructed in terms of its neighbors  $(x_{i1}, \dots, x_{ik})$ . Because weights of  $w_j^i$  are sum up to 1, so eqn.1 can also be written as

$$\min \varepsilon(W) = \sum_{i=1}^N \left| \sum_{j=1}^k w_j^i (x_i - x_{ij}) \right|^2 \quad (2)$$

For a specific  $x_i$

$$\varepsilon^i(W) = \left| \sum_{j=1}^k w_j^i (x_i - x_{ij}) \right|^2 = \sum_{j=1}^k \sum_{m=1}^k w_j^i w_m^i Q_{jm}^i \quad (3)$$

where  $Q_{jm}^i \in R^{k \times k}$ , is the covariance matrix,

$$Q_{jm}^i = (x_i - x_{ij})^T (x_i - x_{im}) \quad (4)$$

Let  $R^i = (Q^i)^{-1}$ . Solving the least squares problem with constraint  $\sum_{j=1}^k w_j^i = 1$  gives:

$$w_{ij} = \frac{\sum_{m=1}^k (Q^i)^{-1}_{jm}}{\sum_{p=1}^k \sum_{q=1}^k (Q^i)^{-1}_{pq}} \quad (5)$$

In practice, a regularization parameter  $r$  will have to be used for  $Q^i$  before inversion:

$$Q^i = Q^i + rI \quad (6)$$

Secondly, the weights  $W$  are fixed and new  $m$ -dimensional vectors  $y_i$  are sought which minimize the criterion:

$$\min \varepsilon(Y) = \sum_{i=1}^n \left| y_i - \sum_{j=1}^k w_{ij} y_{ij} \right|^2 \quad (7)$$

Re-writing eqn.6 gives

$$\min \varepsilon(Y) = \sum_{i=1}^n \sum_{j=1}^n M_{i,j} y_i^T y_j \quad (8)$$

where  $M$  is a  $n \times n$  positive semidefinite matrix found as  $M = (I - W)^T (I - W)$ , and  $Y$  contains the  $y_i$ 's as its columns.

To be able to solve this problem, the covariance matrix of the  $y_i$ 's can be constrained to be identity. Finding  $Y$  then becomes a well-known problem: minimize  $\text{tr}(YMY^T)$  with constraint  $\frac{1}{n}YY^T = I$ . Using Lagrange multipliers and setting the derivative to zero give

$(M - \Lambda)Y^T = 0$ , where  $\Lambda$  is the diagonal Lagrange multiplier matrix. Clearly, this is an eigenvalue problem. All eigenvectors of  $M$  are solutions, but the eigenvectors corresponding to the smallest eigenvalues minimize  $\varepsilon(Y)$ .

The eigenvector with the smallest eigenvalue corresponds to the mean of  $Y$  and can be discarded to enforce  $\sum_{i=1}^n y_i = 0$ . The next  $m$  eigenvectors then give the  $Y$  which minimizes

eqn.8.

### 3. Extended Locally Linear Embedding Algorithm

#### 3.1. Neighbor Selection Based on Tangent Space Distance

From the definition of LLE we can see that the point and its neighbors must lie on or close to a locally linear patch of the manifold. Usually the correlation of data is computed after neighbors have been decided in Euclidean distance, but sometimes the nearest neighbors in Euclidean distance do not lie on the approximate linear curved face, as shown in figure 1,

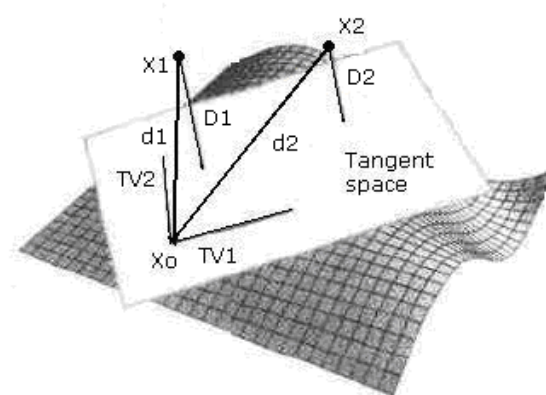


Fig 1. Euclidean distance and tangent space distance from neighbors to point

Where  $X_0$  is the point and lies on a curved face,  $X_1$  and  $X_2$  are its neighbors.  $TV_1$  and  $TV_2$  are tangent vectors of  $X_0$ , and construct the tangent space.  $d_1$ ,  $d_2$  are Euclidean distance from neighbors to  $X_0$ , and  $D_1, D_2$  are distance from neighbors to tangent space of  $X_0$ . According to the definition of LLE we know that shorter  $D$  means better of the locally linear quality of point and its neighbors. From figure 1 we can see that although  $X_1$  is the nearest neighbor of point  $X_0$  in Euclidean distance sense, but in fact  $X_2$  is more suitable to be the nearest neighbor in locally linear sense.

So we introduce tangent space into the algorithm. Firstly, determine tangent space of every point, and then compute distance from neighbors to these tangent spaces, so we can get nearest neighbors which satisfy the hypothesis of a locally linear patch better.

To compute tangent space, we must know differential of the explicit formulation of the function. But in reality problem, there is no explicit formulation of original dataset, but only some discrete samples, so how to compute the tangent space is a question. Fortunately, a step in tangent space alignment method provides a resolvent<sup>[7]</sup>. For every data in higher dimensional space, it can get the approximate tangent space through local PCA, or a set of orthogonal basis.

Suppose  $F$  is a  $d$  dimensional manifold in  $m$  dimensional space mapped by an unknown function  $f(\tau), \tau \in R^d$ , and data  $X = (x_1, x_2, \dots, x_n), x_i \in R^m$  is the image of the unknown function  $x_i = f(\tau_i), i = 1, 2, \dots, n$ . To get the tangent space of  $x_i$ , it is equivalent to get the differential of function  $f(\tau)$  at  $\tau_i$ . Suppose  $f$  is smooth enough, and we can get the first Taylor expansion at some  $\tau$ ,

$$f(\bar{\tau}) = f(\tau) + J_f(\tau) \cdot (\bar{\tau} - \tau) + O(\|\bar{\tau} - \tau\|_2^2) \quad (9)$$

where  $J_f(\tau) \in R^{m \times d}$  is the Jacobi matrix of  $f$  at  $\tau$ .

We record  $f(\tau) = \begin{pmatrix} f_1(\tau) \\ \vdots \\ f_m(\tau) \end{pmatrix}$ , and then  $J_f(\tau) = \begin{pmatrix} \partial f_1 / \partial \tau_1 & \dots & \partial f_1 / \partial \tau_d \\ \vdots & \ddots & \vdots \\ \partial f_m / \partial \tau_1 & \dots & \partial f_m / \partial \tau_d \end{pmatrix}$ , so the tangent

space  $\Gamma_\tau$  of  $f$  at  $\tau$  is expanded by  $d$  column vectors of  $J_f(\tau)$ ,  $\Gamma_\tau = \text{span}(J_f(\tau))$ .

Because we do not know function  $f$ , so we can not compute  $J_f(\tau)$  directly, but we know

it is a  $d$  dimensional space, so if we know a set of orthogonal basis of  $J_f(\tau)$ , we can get  $J_f(\tau) \cdot (\bar{\tau} - \tau) = Q_\tau \theta_\tau^*$ , where  $\theta_\tau^*$  is the coordinate of  $\bar{\tau} - \tau$  at tangent space corresponding to  $Q_\tau$ . Because  $f(\bar{\tau}) - f(\tau) \approx J_f(\tau) \cdot (\bar{\tau} - \tau)$ , so we can get  $f(\bar{\tau}) - f(\tau) \approx Q_\tau \theta_\tau^*$ , that is  $f(\bar{\tau}) \approx Q_\tau \theta_\tau^* + f(\tau)$ .

We need to compute all the tangent space, so for a certain point  $x_i$ , its neighbor  $x_{ij} \approx x_i + Q_i \theta_i$ . Construct followed optimum function,

$$\min_{x_i^*, Q_i, \theta} \sum_{j=1}^k \|x_{ij} - (x_i^* + Q_i \theta_i)\|_2^2 = \min_{x_i^*, Q_i, \theta} \|X_i - (x_i^* e^T + Q_i \theta_i)\|_F^2 \quad (10)$$

Where  $X_i = (x_{i1}, x_{i2}, \dots, x_{ik})$ ,  $Q_i \in R^{m \times d}$ ,  $\theta_i = (\theta_1, \theta_2, \dots, \theta_k) \in R^{d \times k}$ , so  $Q_i$  can be seen as  $d$  approximate orthogonal basis of tangent space of  $x_i$ . The question can be resolved by singular value decomposition (SVD) to  $X_i(I - \frac{1}{k} e e^T)$ , where  $Q_i$  is the  $d$  eigenvectors corresponding to  $d$  maximum eigenvalues of  $X_i(I - \frac{1}{k} e e^T)$ . So we can get every tangent space of each point.

Because the tangent space we get is a set of orthogonal basis but not an analytic expression, so we can not compute the distance directly. Firstly we consider three dimensional condition.

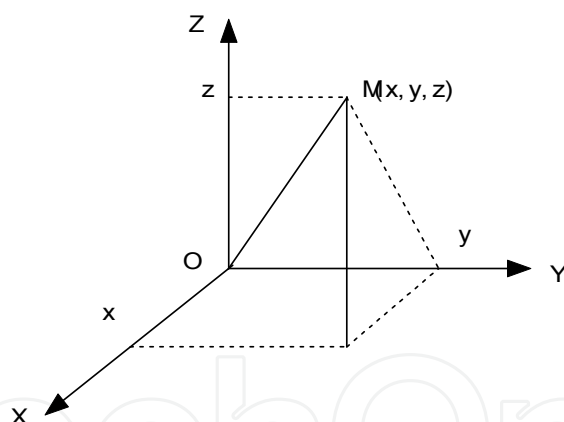


Fig. 2. Distance from point to plain

From figure 2 we can see that, the distance from point  $M$  to plain  $XOY$  is the coordinate of  $M$  in  $Z$  axis, and we extend it to higher dimensional space. Suppose data set is  $m$  dimension, and eigenvectors of  $X_i(I - \frac{1}{k} e e^T)$  is  $E = (e_1, e_2, \dots, e_m)$ , so a set of orthogonal basis of tangent space of point is  $Q = (e_1, e_2, \dots, e_d)$ , then the distance of neighbors to the tangent space can be written as  $\sqrt{x_{d+1}^2 + x_{d+2}^2 + \dots + x_m^2}$ . So we can get tangent space distance dataset to each point, and then nearest neighbors in tangent space can be arrived.

To evaluate the performance of dimension reduction, some researchers propose residual variance according to Input/Output mapping quality, and that is the description

effectiveness of the original data in higher dimensional space. It is defined as  $1 - \rho_{D_x D_y}^2$ , where  $\rho$  is the standard linear correlation coefficient, taken over all entries of  $D_x$  and  $D_y$ , where  $D_x$  and  $D_y$  are the matrices of Euclidean distances (between pairs of points) in the high-dimensional and corresponding low-dimensional spaces, respectively. According to the definition, we can see that the lower the residual variance is, the better high-dimensional data is represented in the embedded space. S-curve dataset is uniform sampled from noiseless three dimensional S-curve curved face. We will reduce the dimension of these two datasets, and compare the effectiveness by residual variance in Euclidean distance and tangent space distance respectively.

S-curve dataset is shown in figure 3. We choose 30 neighbors, and the nearest 30 neighbors of the first point distribution are shown in figure 4. Where the yellow points are neighbors in tangent space distance and green points are neighbors in Euclidean distance, and the red line is the tangent space of the first point. According to the character of S-curve, we can see from the figure that yellow points are distributed in tangent space of the point, and green points are distributed in curve face in relative sense.

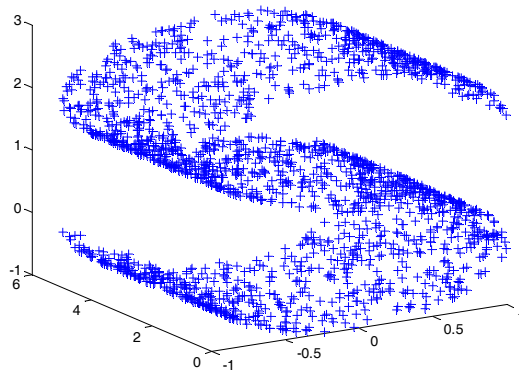


Fig. 3. S-curve dataset

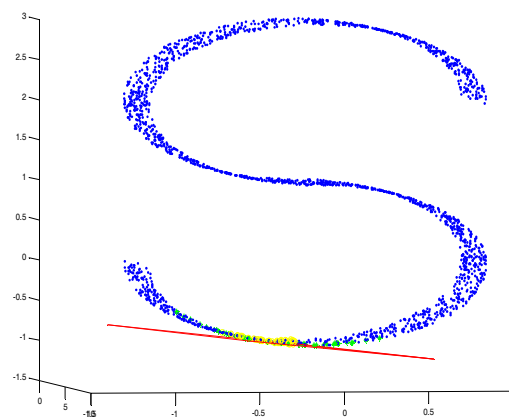


Fig. 4. Neighbors in Euclidean distance and tangent space distance

To test the performance of the algorithm, we compute residual variances under 1 to 50 neighbors, and the result is shown in figure 5,



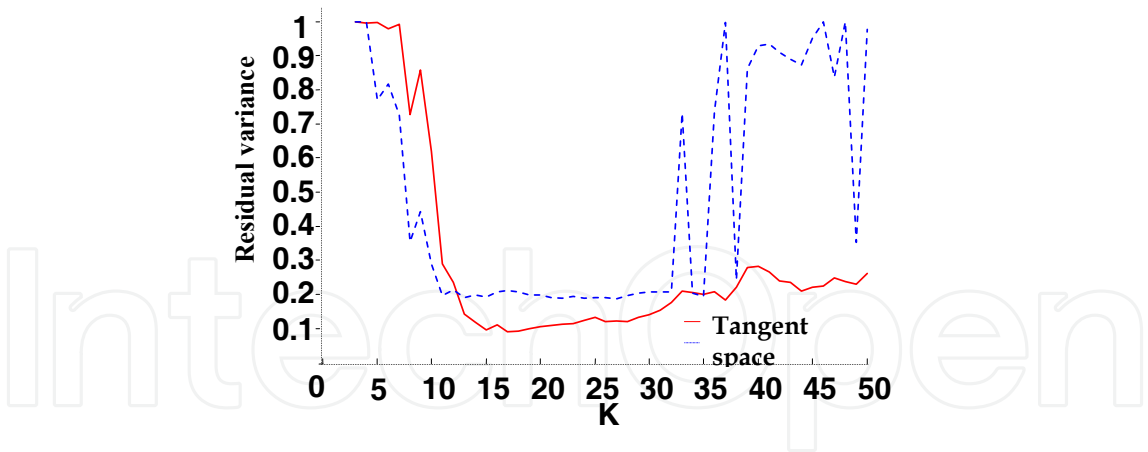


Fig. 5. S-curve dataset residual variance

It can be seen from the figure that when there are few neighbors, the locally structure can not be expressed well, so the residual variance is much big. With more neighbors, performances of two methods both increase, but with more and more neighbors, the method in Euclidean distances can not guarantee the local linear characteristic, so the result is not good as in tangent space distance method obviously.

3.2. Intrinsic Dimensionality Estimation

According LLE algorithm the projection data is the bottom  $d$  nonzero eigenvectors of sparse matrix and  $d$  is the dimension of the feature space. In all the studies conducted by other authors,  $d$  is mostly selected based on the experience, through plenty of tests, which restricts the autonomy and adaptivity of the algorithm to certain extent. Considerable efforts have been invested by other authors into the improvement of the autonomy and adaptivity of the algorithm. For example, Kegl computes the correlation dimension using Grassberger and Procaccia (G-P) algorithm, which is efficient but still suffers from several drawbacks In this paper, we will show certain improvements made to the G-P algorithm using linear fit, which not only realizes the adaptation of the algorithm to a large extent, but also improves the precision of the dimension estimation. According to the concept of topological dimension, the ID of  $X$  is the topological dimension of the support of the distribution of  $X$  [8]. However, when data set is finite, it is difficult to apply the original definition to the ID estimate. The most popular definition, among many proposed, is called correlation dimension, which states that given a data set  $X$  with dimension  $D$  and a distance  $r$ , the number of samples coupled with the distance less than  $r$  is in direct ratio to  $r^D$ . At present, the most frequently used algorithm for estimating the correlation dimension is G-P algorithm, which is summarized as follows:

Given a finite data set  $\Phi = \{x_1, \cdots, x_n\}$  in metric space  $X$ , let

$$C_n(r) = \frac{2}{n(n-1)} \sum_{i=1}^n \sum_{j=i+1}^n H(r - |x_i - x_j|) \tag{11}$$

where  $H(s)$  is an indicator function with  $H(s) = 1$  when  $s > 0$  and  $H(s) = 0$  when  $s < 0$ . For a countable set  $\Phi = \{x_1, \cdots, x_n\}$ , the correlation integral is defined as



$C(r) = \lim_{r \rightarrow 0} C_n(r)$ . If the limit exists, the correlation dimension  $D_{cor}$  of  $\Phi$  is equal to

$$D_{cor} = \lim_{r \rightarrow 0} \frac{\log C(r)}{\log(r)} \quad (12)$$

It has been proven that for a finite sample, the zero limit cannot be achieved. Thus, the original estimate procedure proposed by Grassberger and Procaccia involves plotting  $\log C(r)$  versus  $\log r$  and then measuring the slope of the linear region of the curve<sup>[9][10]</sup>.

The correlation dimension of the finite data set  $\Phi$  can then be expressed as

$$D_{cor} = \frac{\log C(r_2) - \log C(r_1)}{\log(r_2) - \log(r_1)} \quad (13)$$

The algorithm for computing ID can be summarized in the following steps: (a) to select a series of  $r$  with different scales, (b) to compute the corresponding correlation integrals, (c) to identify the linear part of the curve and then to measure the slope of the identified linear region, and (d) to compute the correlation dimension of  $\Phi$ . Among these steps, the selection of the linear part is the most crucial.

Given a countable data set  $\Phi = \{x_1, \dots, x_n\}$  and different scales of distance  $\{r_1, \dots, r_m\}$ , we can obtain a series of corresponding correlation integrals  $\{\log C(r_1), \dots, \log C(r_m)\}$  and plot  $\log C(r)$  versus  $\log r$  curves. The objective is to identify the linear region of this curve. To do so, a third order linear fit is applied to the data series  $(\log C(r_i), \log r_i)$  ( $i = 1 \dots m$ ) to yield:

$$y = a_1 x^3 + a_2 x^2 + a_3 x + a_4 \quad (x \in \log r_i) \quad (14)$$

From Taylor Expansion, in order to make eqn.14 linear, the high order terms must approach to zero, that is,  $f(x) = a_1 x^3 + a_2 x^2 \approx 0$ . Hence, the linear approximation of eqn.13 can be achieved. In other words, the linear part of the curve can be identified and then the correlation dimension can be obtained by computing the slope of linear part of the curve.

Tennessee Eastman (TE) process is a standard test process proposed by J. J. Downs and E. F. Vogel<sup>[11]</sup>. Here we use Tennessee Eastman Process (TEP) data to test the algorithm, which involves selecting a series of  $r$  with different scales and computing the corresponding correlation integrals. TEP has been a well-known benchmark process for comparing various process monitoring methods. The details on the process description can be found in Ref<sup>[12]</sup>. The curve is plotted in figure 6-1. Applying the third order linear fit to the curve yields

$$Y = a_1 (\log r_i)^3 + a_2 (\log r_i)^2 + a_3 (\log r_i) + a_4 \quad (15)$$

Let  $F = a_1 (\log r_i)^3 + a_2 (\log r_i)^2$ , the absolute value of  $F$  is plotted in figure 6-2.

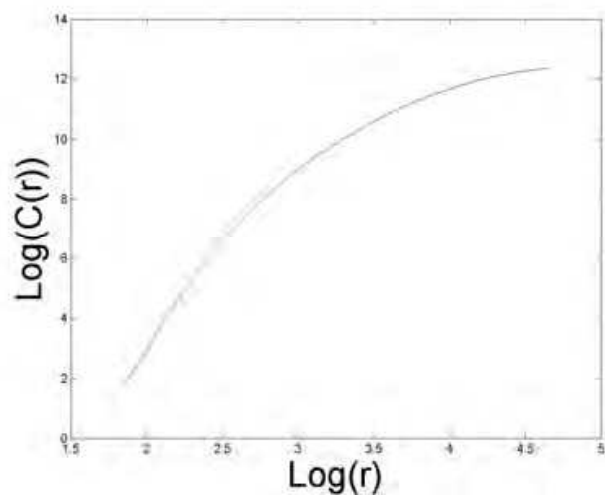


Fig. 6-1.  $\text{Log}C(r) - \log r$  curve

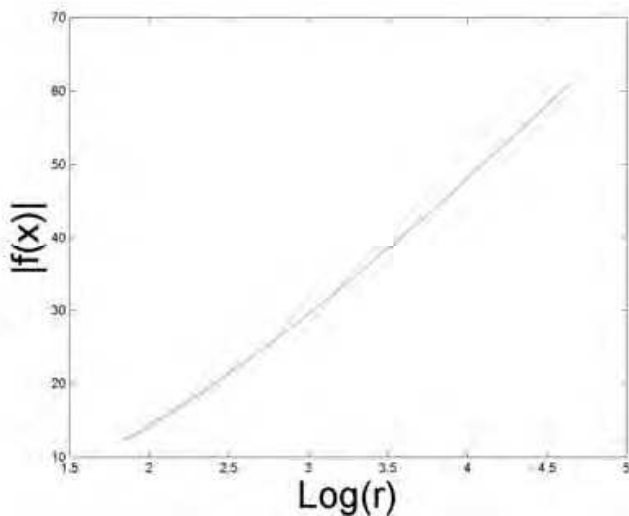


Fig. 6-2. Nonlinear part of fitting curve

Given a threshold  $h$ , the bonds of  $r$  can be obtained. The intrinsic dimension of the data set can be obtained by computing the slope of the linear part of this curve.

4. Feature Extraction Comparison

To test the performance of LLE to feature extraction, we considered the classification problem. Iris data set is a standard database for classification, which consists of three classes, with each class containing  $m=4$  measurements and  $n=50$  observations (see table 1). Here PCA and LLE feature extraction techniques are compared via theoretical and graphical analysis.

The projections of the experimental data onto the first two PCA loading vectors are shown in Figure 7. From which we can see that, separation degree between class 1 and class 2 is small, but aggregation degree of each class is low. And the projections of Iris data by LLE are shown in Figure 8. It can be seen that the separation degree between class 1 and class 2 is much bigger than PCA and the aggregation degree of class 3 is much bigger.

Class 1 : Iris Virginica	Mean	Std. Deviation	Range
Sepal length	6.59	0.64	4.9-7.9
Sepal width	2.98	0.32	2.2-3.8
Petal length	5.55	0.55	4.5-6.9
Petal width	2.03	0.27	1.4-2.5
Class 2 : Iris Versicolor	Mean	Std. Deviation	Range
Sepal length	5.94	0.52	4.9-7.0
Sepal width	2.77	0.31	2.0-3.4
Petal length	4.29	0.47	3.0-5.1
Petal width	1.33	0.20	1.0-1.8
Class 3 : Iris Setosa	Mean	Std. Deviation	Range
Sepal length	5.01	0.35	4.3-5.8
Sepal width	3.43	0.38	2.3-4.4
Petal length	1.46	0.17	1.0-1.9
Petal width	0.30	0.40	0.1-3.0

Table 1. Statistics of Iris data

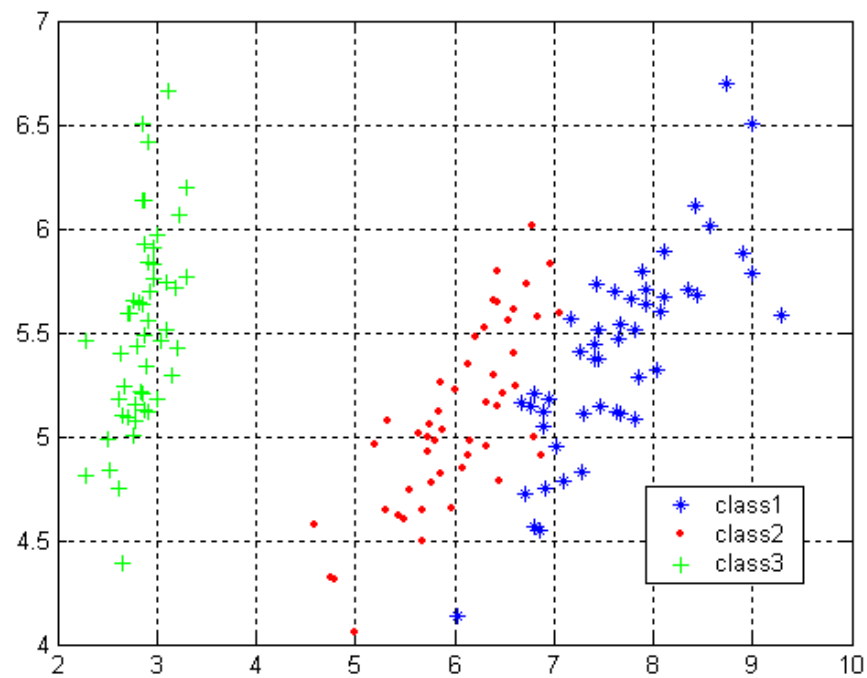


Fig. 7. Projection of Iris data onto the first two PCA loading vectors

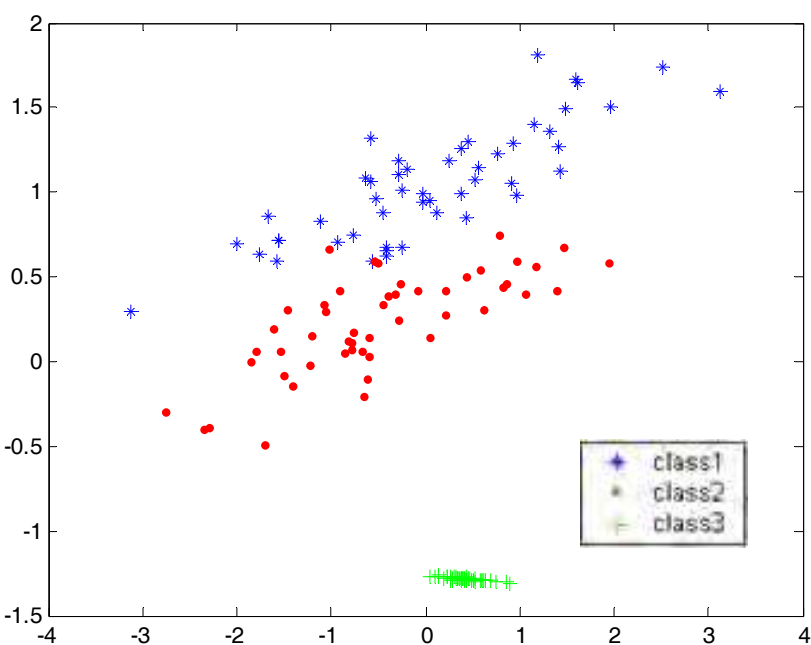


Fig. 8. Two dimension projection of Iris data by LLE

Figure 9 and Figure 10 give the projection of Iris data onto the first three PCA loading vectors and the three dimensional projection by LLE. According to separability criterion, the LLE projection performance of inner-class distance and intra-class distance is much better than PCA. It indicates that LLE algorithm has good description performance and consistency with the data feature.

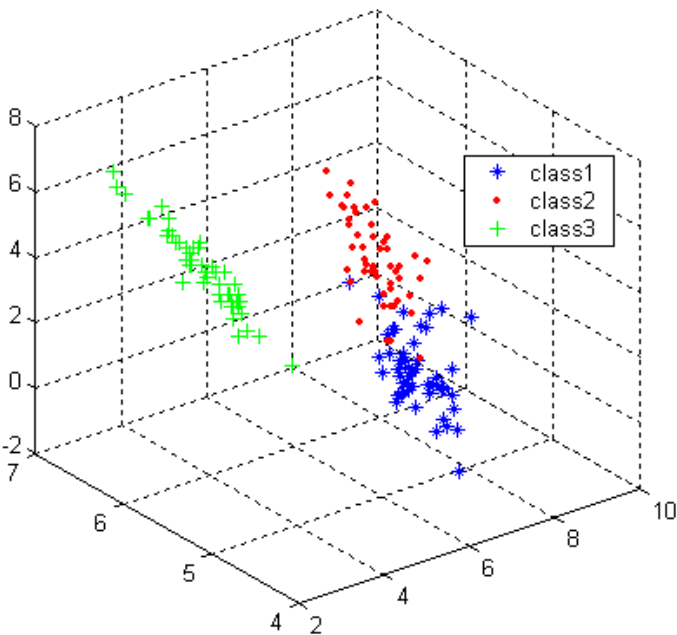


Fig. 9. Projection of Iris data onto the first two PCA loading vectors

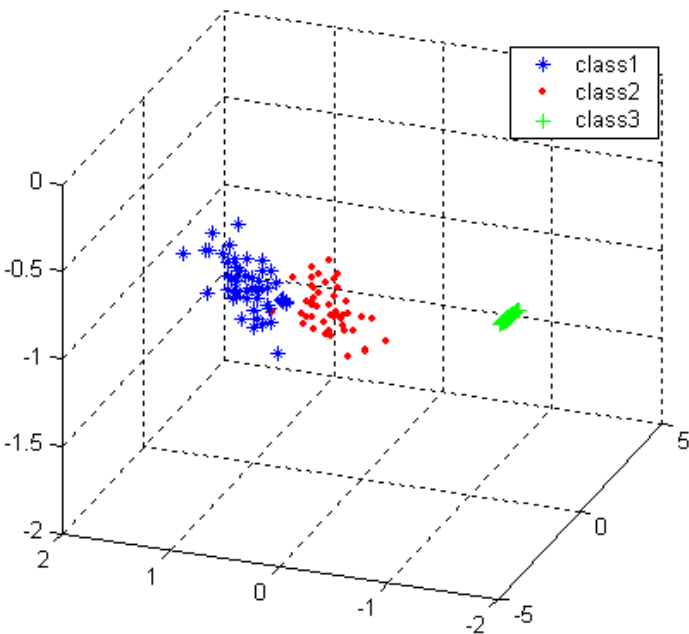


Fig. 10. Three dimensional projections of Iris data by LLE

Table 2 gives the quantitative analysis of inner-class distance and intra-class distance of the two methods, where  $S_w$  is inner-class distance;  $S_b$  is intra-class distance and  $J_d = S_b / S_w$ . The table demonstrates that compared with PCA, LLE can capture the data feature more effectively.

	$S_b$	$S_w$	$J_d$
PCA	11.8152	75.4084	0.157
LLE	2.8005	9.9729	0.281

Table 2. Performance comparison of PCA and LLE

5. Fault detection

LLE cannot compare the projection data with the original data like PCA. But after the projection, LLE can keep the topological structure of the original data as well as the similarity of normal data and illed data. Therefore, we can perform fault detection by computing the inter-class distance between the testing data and the training data[13][14]. Inter-class distance is an important index in pattern recognition. For a multi-class estimation, suppose  $x_k^{(i)}$  and  $x_l^{(j)}$  are vectors of class  $W_i$  and class  $W_j$ , respectively, and  $\delta(x_k^{(i)}, x_l^{(j)})$  is the distance between the two vectors. The average distance between all classes of the vectors is

$$J_d(x) = \frac{1}{2} \sum_{i=1}^c P_i \sum_{j=1}^c P_j \frac{1}{n_i n_j} \sum_{k=1}^{n_i} \sum_{l=1}^{n_j} \delta(x_k^{(i)}, x_l^{(j)}) \tag{16}$$

where  $c$  is the number of classes,  $n_i$  is the number of samples of class  $w_i$ , and  $n_j$  is the number of samples of class  $w_j$ .  $P_i$  and  $P_j$  are prior probabilities of the corresponding classes, respectively. Here, we assume  $P_i = P_j = \frac{1}{c}$ .

After have been projected to the feature space, the distance between the testing data and training data can be computed. Thus, we can get the similarity of the testing data and training data.

Suppose  $x_k = \{x_1, x_2 \cdots x_n\}$  is the training data and  $x_j$  is the testing data. eqn.16 can be simplified as

$$J'_d(x_j) = \frac{1}{2} \sum_{p=1}^2 P_p \sum_{q=1}^2 P_q \frac{1}{n} \sum_{k=1}^n \delta(x_k^{(i)}, x_j)$$

(17)

The fault will be alerted if  $J_d < h$ , where  $h$  is a threshold.

Case	Disturbance	Type
1	<i>A/C</i> feed ratio, <i>B</i> composition constant	Step
2	<i>B</i> composition, <i>A/C</i> ratio constant	Step
3	<i>D</i> feed temperature	Step
4	Reactor cooling water inlet temperature	Step
5	Condenser cooling water inlet temperature	Step
6	<i>A</i> feed loss	Step
7	<i>C</i> header pressure loss – reduced availability	Step
8	<i>A, B, C</i> feed composition	Random variation
9	<i>D</i> feed temperature	Random variation
10	<i>C</i> feed temperature	Random variation
11	Reactor cooling water inlet temperature	Random variation
12	Condenser cooling water inlet temperature	Random variation
13	Reaction kinetics	Slow drift
14	Reactor cooling water valve	Sticking
15	Condenser cooling water valve	Sticking
16–20	Unknown	Unknown
21	The valve for stream 4 was fixed at the steady state position	Constant position

Table 3. Process faults for the Tennessee Eastman process

To validate the fault detection performance of LLE method, Tennessee Eastman process data

is used. A set of programmed faults (Fault 1–21) is listed in Table 3. In addition, one training data set (Fault 0) consisting of 500 samples is generated with no fault. Twenty-one testing data sets corresponding to the twenty-one classes of faults in Table 3 are generated, and each set consists of 960 samples. The fault in each testing data set is introduced from sample 160 and the number of variables is  $m = 52$ .

Figures 11-13 show fault detection results of Tennessee Eastman Process in case of fault 4, fault 10 and fault 16, respectively. The confidence limits are also shown in these figures.

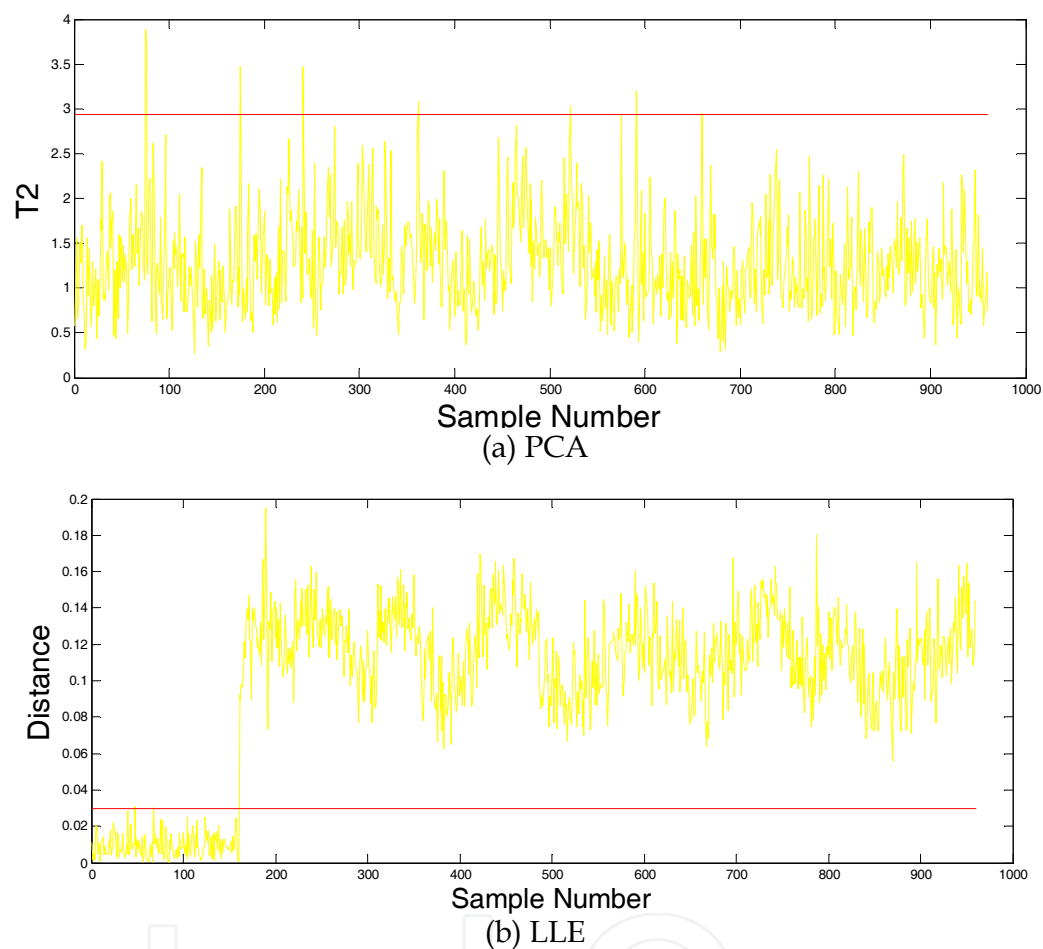
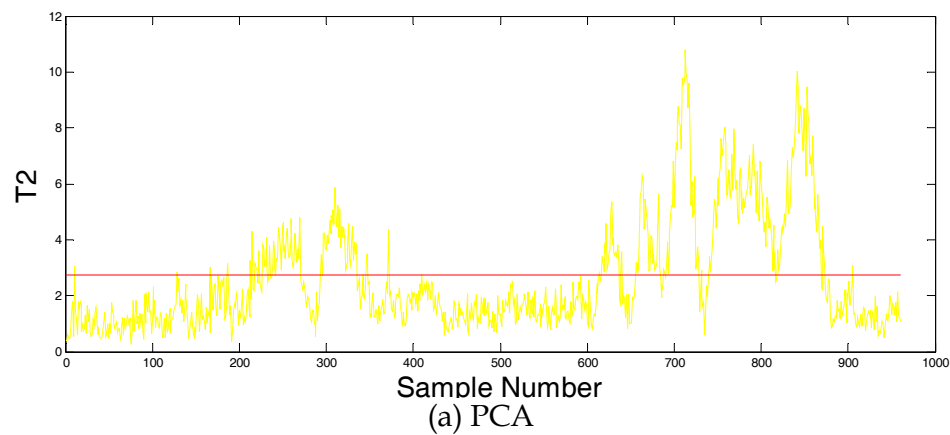


Fig. 11. Fault detection results of Tennessee Eastman process in case of fault 4: (a) PCA and (b) LLE





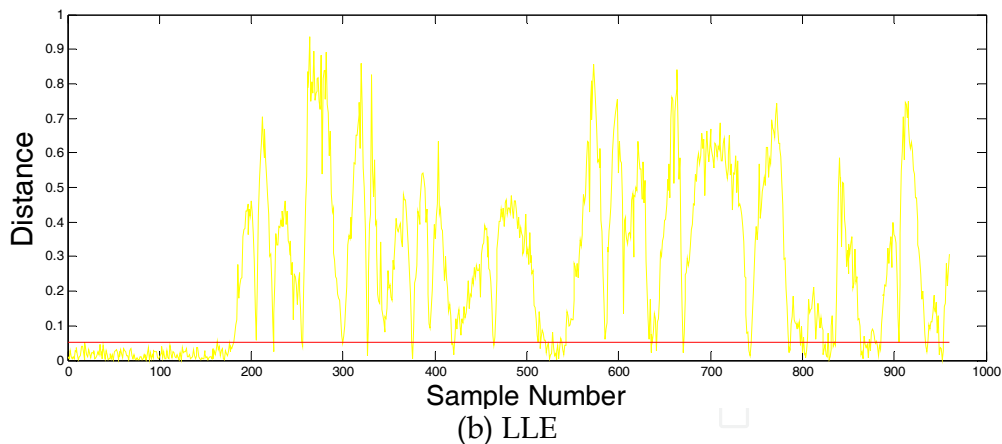


Fig. 12. Fault detection results of Tennessee Eastman process in case of fault 10: (a) PCA and (b) LLE

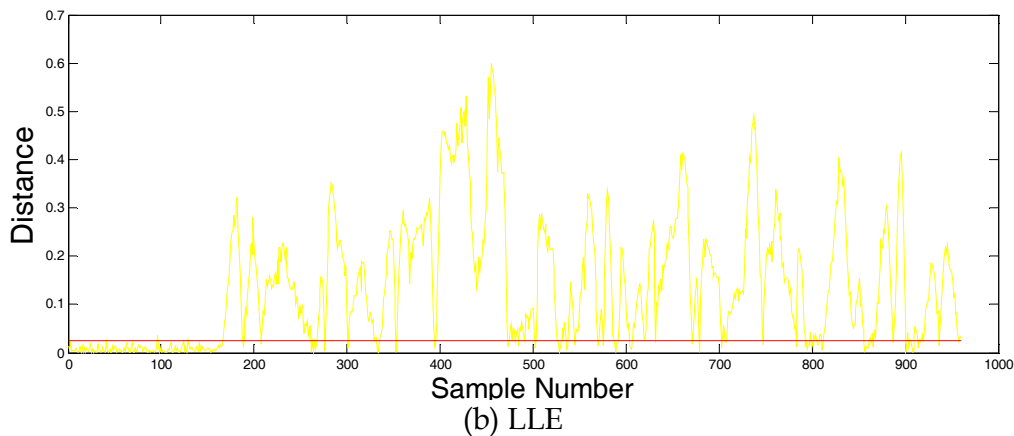
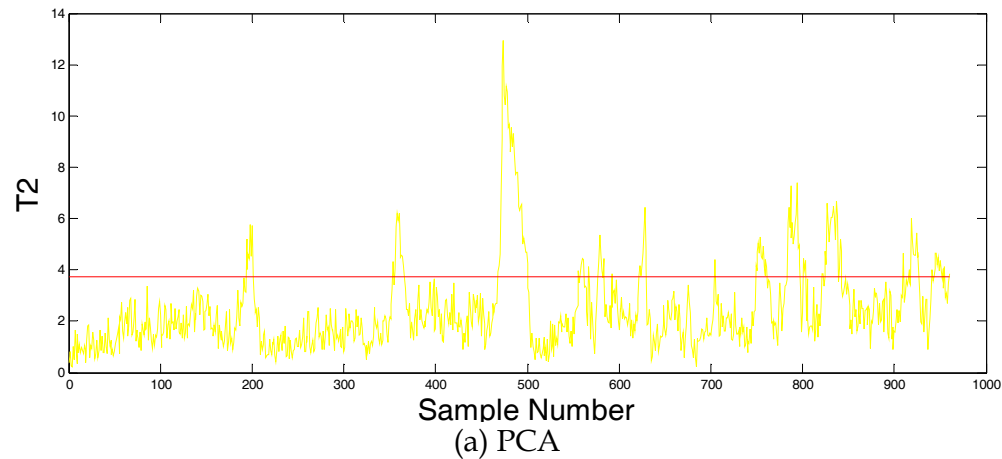


Fig. 13. Fault detection results of Tennessee Eastman process in case of fault 16: (a) PCA and (b) LLE

The false positive(FP) rate and false negative(FN) rate of figures 11-13 are listed in table 4. From table 4, it can be concluded that the FP rate of LLE is a little higher than PCA, which is mostly because the PCA is insensitivity to the change of data. But the FN rate of LLE is much lower than PCA, which is 4.498% in average. On the other hand, the FN rate of PCA achieves 77.53% in average, which means that the approach based on LLE can detect each

type of the faults, but PCA fails to alert these faults.

		False Positive	False Negative
Fault 4	PCA	1 . 25 %	99 . 25%
	LLE	8 . 75%	0
Fault 10	PCA	1 . 25%	60 . 25%
	LLE	7 . 75%	7 . 75%
Fault 16	PCA	0	81 . 75%
	LLE	5 . 63%	7 . 62%

Table 4. FN and FP of fault 4、 10 and 16

6. References

[1] PEARSON K. On lines and planes of closest fit to systems of points in space[J]. Philos Mag, 1901, 6(2):559 - 572.

[2] SAMUEL A L. Some studies in machine learning using the game of checkers[J]. IBM Journal Research and Development, 1967,11(4): 601-617.

[3] DUNIA R, QIN S Joe, EDGAR T F, MCAVOY T J. Use of principal component analysis for sensor fault identification[J]. Computers chem. Eng, 1996, 20(7):13-718

[4] JACKSON J E. Principal component and factor analysis: Part I - principal component[J]. Journal of Quality Technology, 1980,12:201-213

[5] HASTIE T. Principle curves and surfaces[R]. Technical Report, Laboratory for Computational Statistics, Stanford University,1984

[6] ROWEIS S T, SAUL L K. Nonlinear dimensionality reduction by locally linear embedding[J]. Science, 2000, 290(22): 2323-2326

[7] Zhang Zhen-yue, Zha Hong-yuan. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment[J]. SIAM J.Sci. Comput. 2004, 26(1): 313- 338.

[8] KEGL B. Intrinsic dimension estimation using packing numbers[C].//Neural Information Processing Systems 15 (NIPS'2002), 2002.

[9] BELUSSIA A, FALOUTSOS C. Spatial join selectivity estimation using fractal concept[J]. ACM Transaction on Information Systems, 1998, 16(2): 161-201.

[10] HASTAD J. Clique is hard to approximate within  $n^{1-\epsilon}$  [C].// Proceedings of the 37th Annual Symposium on Foundations of Computer Science FOCS'96,1996,627-636.

[11] DOWNS J J, VOGEL E F. Plant-wide industrial process control problem[J]. Computers&Chemical Engineering. 1993, 17(3): 245-255.

[12] Tracy, N. D., Young, J. C., & Mason, R. L. Multivariate control charts for individual observations. Journal of Quality Technology, 1992, 24:88-95.

[13] MARQUES J P. Pattern Recognition: Concepts, Methods and Applications[M]. Berlin: Springer, 2001.

[14] DEVIJVER P A, KITTLER J. Pattern recognition theory and applications[M]. Berlin: Springer-Verlag, 1987.



## **Fault Detection**

Edited by Wei Zhang

ISBN 978-953-307-037-7

Hard cover, 504 pages

**Publisher** InTech

**Published online** 01, March, 2010

**Published in print edition** March, 2010

In this book, a number of innovative fault diagnosis algorithms in recently years are introduced. These methods can detect failures of various types of system effectively, and with a relatively high significance.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Wei Zhang and Xu Yang (2010). Fault Detection Method Based on Extended Locally Linear Embedding, Fault Detection, Wei Zhang (Ed.), ISBN: 978-953-307-037-7, InTech, Available from:

<http://www.intechopen.com/books/fault-detection/fault-detection-method-based-on-extended-locally-linear-embedding>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen