

# 改进 LNS 和邻域保持嵌入算法的研究

李元 黄莹莹

(沈阳化工大学信息工程学院 辽宁 沈阳 110142)

**摘要** 传统邻域保持嵌入算法(Neighbor Preserving Embedding, NPE)对具有多中心、方差差异明显特性的高维数据的降维处理效果并不好,因此提出一种改进 LNS 和邻域保持嵌入算法(Modified Local Neighbor Standardization-Neighbor Preserving Embedding, MLNS-NPE),并应用于故障诊断中。利用 MLNS 算法对数据进行处理,对处理后的数据进行 NPE 算法建模。在数值例子和青霉素发酵过程中应用该算法与传统 NPE 算法、核邻域保持嵌入算法(KNPE)、KNN 算法比较,结果表明,采用该算法后,数据多中心和模态差异消除,为后续 NPE 算法的应用提供先决条件,同时相比其他算法故障检测率最高,提高了 NPE 算法对多模态数据的检测能力。

**关键词** 改进 LNS 算法 邻域保持嵌入算法 青霉素发酵 多模态 故障检测

中图分类号 TP2

文献标志码 A

DOI:10.3969/j.issn.1000-386x.2021.02.041

## MODIFIED LNS AND NEIGHBORHOOD PRESERVING EMBEDDING ALGORITHM

Li Yuan Huang Yingying

(College of Information Engineering, Shenyang University of Chemical Technology, Shenyang 110142, Liaoning, China)

**Abstract** Traditional neighbor preserving embedding (NPE) is not effective in dimensionality reduction of high-dimensional data with multi-center and obvious variance differences. Therefore, an improved LNS and neighbor preserving embedding (MLNS-NPE) algorithm is proposed and applied to fault diagnosis. It used MLNS algorithm to process the data, then carried out NPE algorithm modeling on the processed data. In the process of numerical examples and penicillin fermentation, the comparison results between the algorithm and traditional NPE algorithm, kernel neighborhood preserving embedding algorithm (KNPE) and KNN algorithm show that after the algorithm is adopted, the data multicenter and modal difference are eliminated, which provides a prerequisite for the application of subsequent NPE algorithm. At the same time, compared with other algorithms, the fault detection rate is the highest, and the detection capability of NPE algorithm for multimodal data is improved.

**Keywords** Modified LNS algorithm Neighborhood preserving embedding algorithm Penicillin fermentation Multimodal Fault detect

## 0 引言

随着自动化技术日新月异的发展和人民生活水平的不断提高,人们对生产产品的质量要求也越来越高,因此生产过程也趋于复杂化。生产过程的复杂化不可避免地产生产大量且维数较高的数据,故障诊断技术就

是利用这些可以被采集、存储、分析和利用的过程数据进行数据建模,进而进行故障诊断。但现实的是在建模时,这些高维数据通常计算量较大并且很难建立监控模型,因而有必要对这些数据进行降维处理。降维后的数据由于包含原始数据大部分特征并且容易进行监控,故而受到很多专家学者的青睐。目前常见的数据降维算法有很多,比较熟知的线性降维技术有主元

分析(Principal Component Analysis, PCA)<sup>[1]</sup>、偏最小二乘(Partial Least Squares, PLS)等<sup>[2]</sup>,非线性降维技术包括局部线性嵌入(Locally Linear Embedding, LLE)<sup>[3]</sup>、拉普拉斯特征映射(Laplacian Eigenmap, LE)<sup>[4]</sup>等。上述降维方法所能处理的数据类型和动机都不同,因而彼此的目标函数也有些许差异。PCA 是基于数据驱动方法的多元统计过程控制中常用的降维方法<sup>[5]</sup>,它寻找数据分布方差最大的方向,并将该方向作为第一主元方向。PLS 可以用于提取与质量变化相关的主元。LLE、LE 是近年来比较热门的非线性降维技术,它们的共同点在于都是利用样本点的局部近邻信息来获知其内部结构信息,但它们有一个缺点就是“样本外”问题<sup>[6]</sup>。所谓“样本外”问题是指它们利用训练数据建立过程模型,但是新来的测试数据(训练数据之外)不能直接投影到过程模型上而是需要重新计算,因此给计算带来困难。He 等<sup>[7]</sup>提出了一种新的流形学习算法-邻域保持嵌入(Neighbor Preserving Embedding, NPE)算法,NPE 算法虽然由 LLE 算法延伸而来,但与非线性降维技术 LLE 不同的是,NPE 是一种线性降维技术,它除了继承了 LLE 算法的邻域结构保持特性还有效解决了“样本外”问题,它和 LLE 算法一样都是利用训练数据建立模型,但新来测试数据可通过一个投影矩阵直接投影到过程模型上。

由于 NPE 算法的简便性并且降维后原高维空间线性结构依然可以被保持,因此被广泛应用于人脸识别和故障诊断中。Liu 等<sup>[8]</sup>提出了一种新的特征提取方法-联合稀疏邻域保持嵌入(JSNPE)。NPE 可以保持局部结构但却忽略了全局特征,因此赵小强等<sup>[9]</sup>提出了张量全局-局部邻域保持嵌入方法(TGNPE)。针对动态过程,杨健等<sup>[10]</sup>提出了时序约束 NPE 算法(TCNPE)。针对高维分类数据,寇勃晨等<sup>[11]</sup>提出了自适应 NPE 方法(IM-ANPE)。针对多模态数据,宋冰等<sup>[12]</sup>提出了 LSNPE 算法。

NPE 算法使用 z-score 方法标准化,z-score 方法可以消除量纲但当数据疏密程度不同时标准化后数据的多模态特征依然明显。因此针对方差差异明显的多模态数据,引入 Ma 等<sup>[13]</sup>提出的局部近邻标准化(Local Neighbor Standardization, LNS)方法,该方法在消除量纲的同时可以使数据中心平移到原点,且标准化后的数据疏密程度相同。

综上所述,针对现代生产过程数据多模态和复杂数据分布问题,本文提出 MLNS-NPE 算法对工业过程进行故障诊断。MLNS 方法提高了 NPE 算法对稀疏多模态数据的故障检测性能,NPE 算法在数据降维的同

时提取数据的局部信息,削弱原始空间离群点的影响。最后将该方法用于故障诊断中,取得良好的故障检测效果。

## 1 邻域保持嵌入(NPE)算法

NPE 是流形算法的一种,为避免“样本外”数据问题,它的目标就是寻找一投影矩阵  $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_d)$  将原始数据集  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \in \mathbf{R}^D$  映射到特征空间中,特征空间的数据集  $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n) \in \mathbf{R}^d (d < D)$ 。过程如下:

(1) 构造邻接图:使用 k 近邻法,若  $x_j$  属于样本  $x_i$  的 k 近邻,则两者之间有一条直接的边。

(2) 构造权重矩阵:通过最小化重构误差函数得到局部权重值,重构误差函数为:

$$\Phi(\mathbf{W}) = \min \sum_{i=1}^n \|\mathbf{x}_i - \sum_{j=1}^n \mathbf{w}_{ij} \mathbf{x}_j\|_2^2 \quad (1)$$

式中: $\mathbf{w}_{ij}$  为样本  $x_i$  与其近邻  $x_j$  之间的权重系数; $\mathbf{W}$  为权重矩阵且约束条件  $\sum \mathbf{w}_{ij} = 1$ 。当  $\mathbf{x}_j \in \mathbf{x}_k$  时, $\mathbf{w}_{ij} \neq 0$ ; 当  $\mathbf{x}_j \notin \mathbf{x}_k$  时, $\mathbf{w}_{ij} = 0$ 。

式(1)的求解可以引入拉格朗日乘法,式(1)可以写成:

$$\begin{aligned} \Phi(\mathbf{W}) &= \min \sum_{i=1}^n \|\mathbf{x}_i - \sum_{j=1}^n \mathbf{w}_{ij} \mathbf{x}_j\|_2^2 = \\ &= \min \sum_{i=1}^n \|[x_i - \mathbf{x}_1, x_i - \mathbf{x}_2, \dots, x_i - \mathbf{x}_k]^T \mathbf{w}_i\|_2^2 = \\ &= \min \sum_{i=1}^n \mathbf{w}_i^T \mathbf{z}_i \mathbf{w}_i \quad (2) \\ &\text{s. t. } \sum_{j=1}^n \mathbf{w}_{ij} = 1 \rightarrow \mathbf{w}_i^T \mathbf{o}_k = 1 \end{aligned}$$

式中: $\mathbf{z}_i = (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T$ 。构造拉格朗日函数:

$$L(\mathbf{W}) = \sum_{i=1}^n (\mathbf{w}_i^T \mathbf{z}_i \mathbf{w}_i + \lambda_1 (\mathbf{w}_i^T \mathbf{o}_k - 1)) \quad (3)$$

对  $\mathbf{w}_i$  求导得到式(4),其中  $\lambda = -\frac{1}{2} \lambda_1$ ,  $\mathbf{o}_k$  为  $k$  维的全一向量。

$$2\mathbf{z}_i \mathbf{w}_i + \lambda_1 \mathbf{o}_k = 0 \rightarrow \mathbf{w}_i = \lambda \mathbf{z}_i^{-1} \mathbf{o}_k \quad (4)$$

利用约束条件  $\mathbf{w}_i^T \mathbf{o}_k = 1$  对  $\mathbf{w}_i$  进行归一化后得到:

$$\mathbf{w}_i = \frac{\mathbf{z}_i^{-1} \mathbf{o}_k}{\mathbf{o}_k^T \mathbf{z}_i^{-1} \mathbf{o}_k} \quad (5)$$

因此得到的权重矩阵为:

$$\mathbf{W} = \sum_{i=1}^n \mathbf{w}_i \quad (6)$$

(3) 构造投影矩阵。NPE 的思想是若在原始高维空间权值  $\mathbf{w}_{ij}$  可以线性表示样本  $x_i$ ,那么它同样可以在

投影空间线性表示  $y_i$ , 以此表示其线性结构得到了保持。投影矩阵  $A$  可以通过下式得到。

$$\Phi(A) = \min \sum_{i=1}^n \|y_i - \sum_{j=1}^n w_{ij} y_j\|_2^2 \quad (7)$$

约束条件:  $y^T y = 1 \rightarrow a^T X X^T a = 1$ , 式(7)可以用向

量的形式表示为  $y_i - \sum_{j=1}^n w_{ij} y_j$ , 则:

$$q = y - W y = (I - W) y \quad (8)$$

$$q^2 = q^T q = y^T (I - W)^T (I - W) y = a^T X (I - W)^T (I - W) X^T a \quad (9)$$

构造拉格朗日函数:

$$L(a) = a^T X (I - W)^T (I - W) X^T a + \lambda_2 (a^T X X^T a - 1) \quad (10)$$

对  $a$  求导, 得到式(11), 其中  $\lambda = -\lambda_2$ 。

$$2X(I - W)^T (I - W) X^T a + 2\lambda_2 X X^T a = 0 \quad (11)$$

因此式(7)的最优化问题转化成为  $X M X^T a = \lambda X X^T a$  的广义特征值求解问题。 $a_0, a_1, \dots, a_{d-1}$  是特征值  $\lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_{d-1}$  对应的特征向量,  $M = (I - W)^T (I - W)$ , 则投影矩阵  $A = [a_0, a_1, \dots, a_{d-1}]$ 。

得到投影矩阵  $A$  后, 原始数据被分为特征空间  $Y$  和残差空间  $E$ :

$$X = A Y + E \quad (12)$$

$$Y = (A^T A)^{-1} A^T X = A^T X \quad (13)$$

$$E = X - A Y \quad (14)$$

分别在特征空间和残差空间建立  $T^2$  统计量和  $SPE$  统计量:

$$T^2 = y_i^T A^{-1} y_i = x_i^T A \times \left( \frac{Y Y^T}{n-1} \right)^{-1} \times A^T x_j \quad (15)$$

$$SPE = x_i^T \times (I - A A^T)^T (I - A A^T) \times x_i \quad (16)$$

式中:  $A$  为对角矩阵。

## 2 算法设计

### 2.1 局部近邻标准化

使用上述 NPE 算法时所用的数据标准化方法为 z-score 标准化方法, z-score 方法是针对数据集整体的均值和标准差进行标准化, 它可以实现单一模态数据的标准化处理, 但在处理方差差异明显的多模态数据时, 只能消除量纲, 各模态疏密程度并没有变化。针对方差差异明显的多模态数据, 近年来出现了种新的数据标准化方法即局部近邻标准化(LNS)方法, LNS 标准化方法如下:

$$\hat{x}_i = \frac{x_i - \text{mean}(N(x_i))}{\text{std}(N(x_i))} \quad (17)$$

式中:  $N(x_i)$ 、 $\text{mean}(N(x_i))$ 、 $\text{std}(N(x_i))$  分别为样本  $x_i$  的近邻集、均值和标准差。

LNS 寻找样本  $x_i$  的  $k$  近邻作为近邻集, 对各自模态正常样本而言, 其近邻集的均值和标准差都源于各自模态, 正常样本与其近邻集均值的距离较小, 因此  $x_i - \text{mean}(N(x_i))$  可以使数据中心平移到原点, 相比稀疏模态, 密集模态更靠近原点。稀疏模态方差较大,  $x_i - \text{mean}(N(x_i))$  除以  $\text{std}(N(x_i))$  可以使稀疏模态向内聚拢, 密集模态方差较小, 作商使密集模态数据向外扩散, 因此两个模态经 LNS 处理后多中心变为一个中心并且疏密程度大致相同。正常样本与其近邻集均值的距离小于故障样本与其近邻集均值的距离, 因此 LNS 方法处理后故障样本偏离原点, 远离正常样本。

LNS 虽然能有效处理稀疏多模态数据, 但当故障样本位于多个模态中间时, 它的  $k$  个近邻样本可能同时位于几个不同模态中<sup>[14]</sup>, 此时故障样本近邻集方差波动大, 因此 LNS 方法处理后得到的新样本同正常样本一样接近原点, 此时故障样本混入正常样本导致 LNS 无法识别该故障。采用 MLNS 方法来解决样本近邻间跨模态问题。MLNS 标准化方式如下:

$$\hat{x}_i^1 = \frac{x_i - \text{mean}(N(x_i^1))}{\text{std}(N(x_i^1))} \quad (18)$$

式中:  $N(x_i^1) = \{x_{i1}^1, x_{i2}^1, \dots, x_{i3}^1\}$ 、 $\text{mean}(N(x_i^1))$ 、 $\text{std}(N(x_i^1))$  分别为样本  $x_i$  的第一近邻的近邻集、均值、标准差。

MLNS 方法寻找样本第一近邻的近邻集, 对于正常样本和非位于模态之间的故障样本, 其作用同 LNS 方法。对于位于模态之间的故障样本而言, 无论其第一近邻属于哪个模态, 其近邻集都全部属于该模态, 均值和标准差也都属于该模态, 此时近邻集方差波动较小, 因此经 MLNS 方法处理后, 该故障样本远离数据中心, 偏离正常样本。下面用一数值例子说明: 随机生成一多模态数值例子, 两个模态满足:

$$\text{模态 1: } \begin{cases} s1 \sim N(1, 1) \\ s2 \sim N(-15, 1) \\ s3 \sim N(-15, 1) \end{cases}$$

$$\text{模态 2: } \begin{cases} s1 \sim N(6, 6) \\ s2 \sim N(20, 6) \\ s3 \sim N(20, 6) \end{cases}$$

每个模态选取 300 个训练样本, 50 个测试样本, 共 600 个训练样本, 100 个测试样本。并设置 5 个故障, 分别为  $f1(-40, 45, 70)$ 、 $f2(-30, -50, -16)$ 、 $f3(0, 5, 6)$ 、 $f4(-25, -15, -25)$ 、 $f5(75, 0, 20)$ 。

如图 1 所示, 原始数据包含数据中心不重合的两

个模态:模态 1 密集,模态 2 稀疏。 $f_1$ 、 $f_5$  为稀疏模态故障,  $f_2$ 、 $f_4$  为密集模态故障,  $f_4$  为密集模态的微弱故障,  $f_3$  位于两个模态中间。原始数据经 z-score 标准化处理后消除了量纲,但数据分布仍为稀疏多模态。如图 2 所示,使用 LNS 方法处理后稀疏多模态数据变为疏密程度大致相同的单一模态数据,数据中心平移到原点附近。此时  $f_1$ 、 $f_2$ 、 $f_4$ 、 $f_5$  偏离正常样本,但位于模态间的  $f_3$  混入正常样本,被正常样本覆盖。使用 MLNS 方法处理后稀疏多模态数据变为疏密程度大致相同的单一模态数据,多模态数据中心重合并且平移到原点附近,多模态特性消除,  $f_1$ 、 $f_2$ 、 $f_4$ 、 $f_5$  偏离正常样本的同时  $f_3$  远离正常样本点。

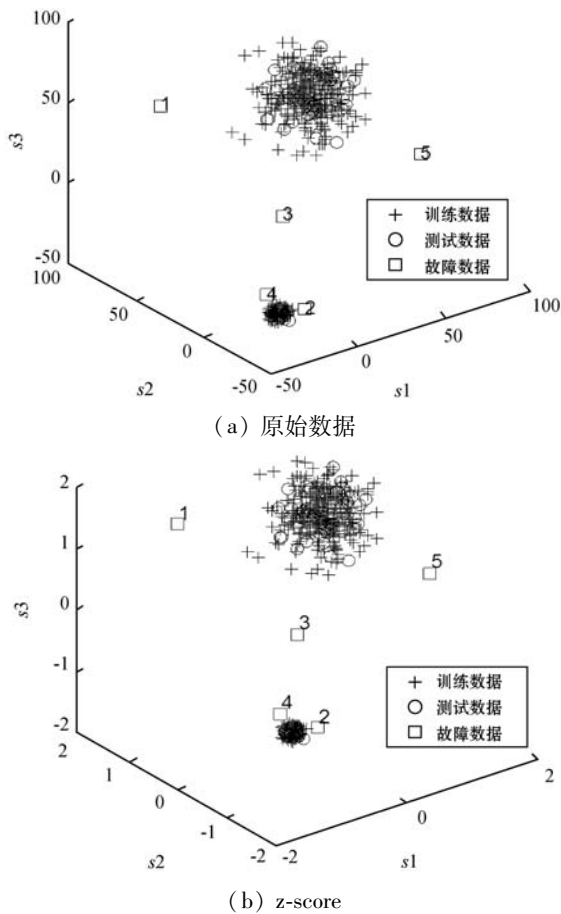


图 1 原始数据分布散点图和 z-score 标准化后散点图

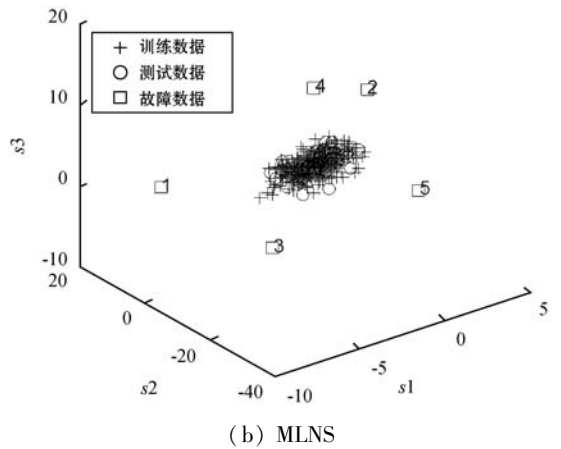


图 2 LNS 和 MLNS 标准化处理后散点图

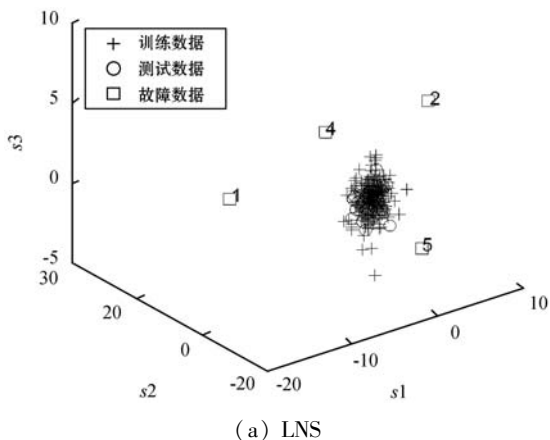
表 1 为  $f_3$  的前 13 近邻,  $f_{3_1}$  表示故障样本  $f_3$  的前 13 近邻,可以看出第 9、10、11、12 个近邻属于第一模态,其余为第二模态,因此  $f_3$  存在近邻样本跨模态问题。 $f_{3_2}$  表示  $f_3$  第一近邻的前 13 个近邻,可以看出,第一近邻属于第二模态,第一近邻的前 13 个近邻全部属于第二模态,因此采用 MLNS 方法避免了模态中心故障近邻集跨模态问题,  $f_3$  被成功检测,提高了 NPE 算法对疏密不同数据的检测能力。

表 1 故障样本  $f_3$  的前 13 近邻

序号	$f_{3_1}$	$f_{3_2}$
1	393	393
2	557	490
3	347	445
4	470	546
5	442	380
6	571	457
7	453	585
8	490	511
9	244	367
10	208	520
11	182	470
12	96	352
13	445	412

## 2.2 故障检测

传统 NPE 算法使用 z-score 标准化方法,但对于多模态数据而言,经 z-score 标准化后,数据仍具有多个中心,即仍然是多模态结构。因此 NPE 算法在处理具有多个中心且疏密程度不同的多模态数据时,其处理效果并不好。针对疏密程度不同的多模态数据的故障检测,本文提出一种基于 MLNS-NPE 的故障检测方法,使用 MLNS 方法代替原始 NPE 算法的 z-score 标准化方法,去除数据的多中心和方差差异特性,然后进行



NPE 建模。基于 MLNS-NPE 的多模态故障检测分为模型建立和在线检测两个阶段：

1) 正常状态下的模型建立：

(1) 收集正常工况下的历史数据集  $X$ , 找到训练样本  $x_i$  第一近邻的局部近邻集  $N(x_i^1)$ , 计算其均值和标准差, 标准化该训练样本。

(2) 在高维空间构造邻接图, 根据式 (6) 构造权重矩阵。

(3) 根据式 (7) 构建投影矩阵。

(4) 计算  $SPE$ 、 $T^2$  统计量。

(5) 利用核密度估计法 (Kernel Density Estimation, KDE)<sup>[15]</sup> 确定  $SPE$  和  $T^2$  控制限  $SPE_\alpha$  和  $T_\alpha^2$ 。

2) 在线检测：

(1) 应用 MLNS 算法标准化新样本  $V$ 。

(2) 将标准化后的测试样本投影到 NPE 模型上, 计算其  $SPE$  和  $T^2$  统计量  $SPE_i$  和  $T_i^2$ 。

(3) 将  $SPE_i$  与  $SPE_\alpha$  比较: 若  $SPE_i < SPE_\alpha$ , 则正常, 否则故障; 将  $T_i^2$  与  $T_\alpha^2$  比较: 若  $T_i^2 < T_\alpha^2$ , 则正常, 否则故障。

2.3 算法流程

MLNS-NPE 算法流程如图 3 所示。

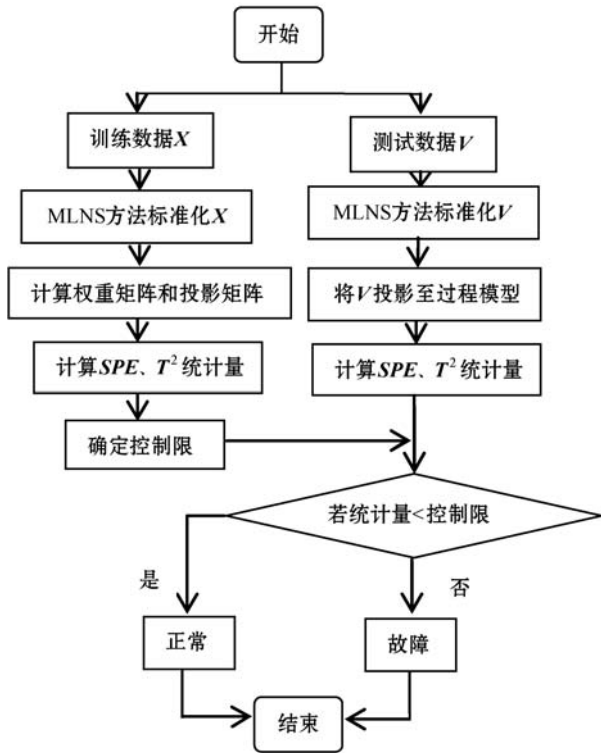


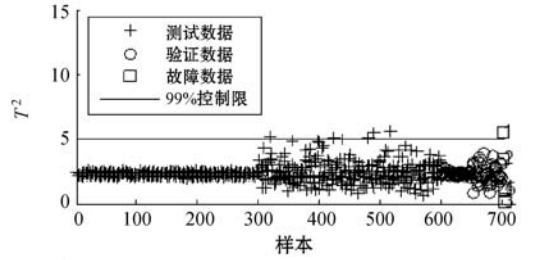
图 3 MLNS-NPE 流程图

3 仿真分析

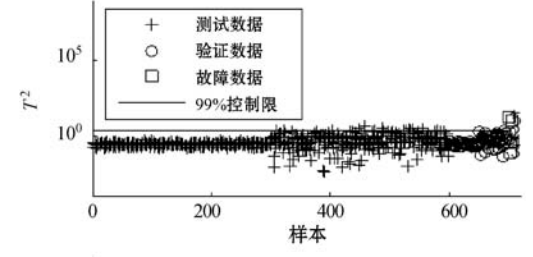
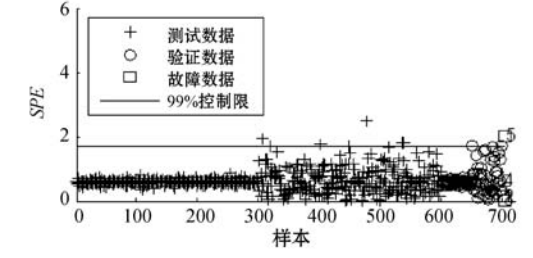
3.1 数值例子

本节数值例子使用上节的数值实例, 分别用 NPE、

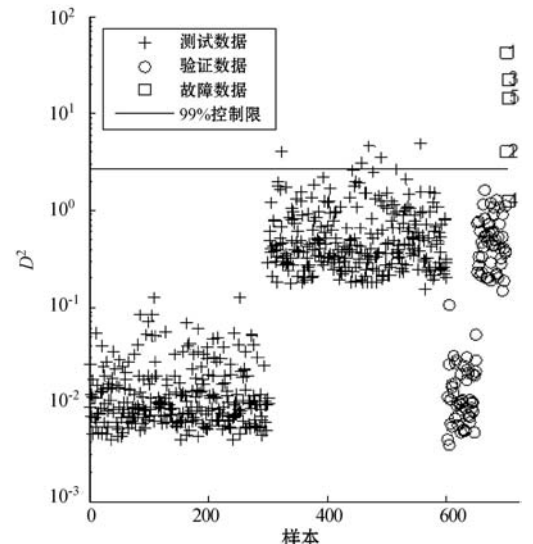
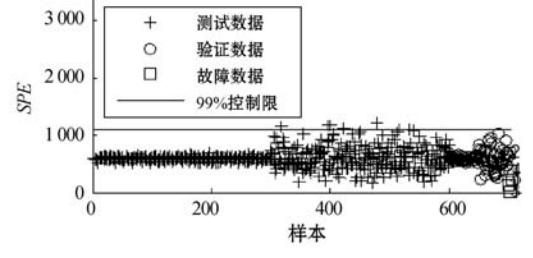
KNPE、KNN、MLNS-NPE 四种算法对该多模态过程进行检测。其中 NPE、KNPE、MLNS-NPE 算法的重构近邻数为 6, KNN 算法的近邻数为 13, MLNS-NPE 算法中第一近邻的 k 近邻数为 13。检测结果如图 4 所示。



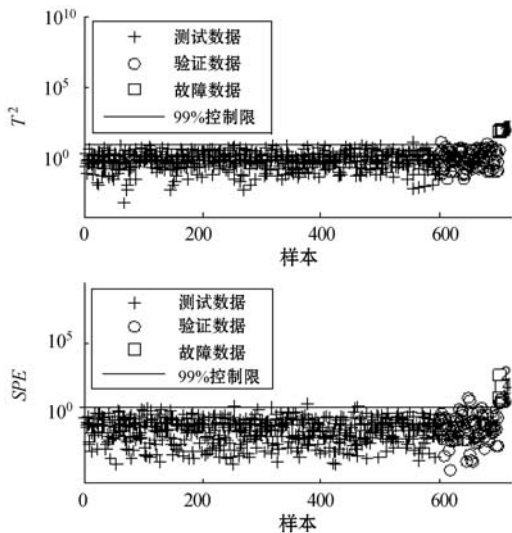
(a) NPE 算法



(b) KNPE 算法



(c) KNN 算法



(d) MLNS-NPE 算法

图 4 四种算法的故障检测结果

使用上述四种方法检测:其中 NPE 和 KNPE 两种方法都使用 z-score 方法进行标准化,检测效果如图 4(a)和图 4(b)所示。NPE 方法的  $SPE$ 、 $T^2$  统计量各检测到一个故障,此时控制限完全由较稀疏模态确定。使用 KNPE 方法,将数据映射到高维,有效降低了非线性性的影响, $T^2$  统计量检测到故障 f1、f5, $SPE$  统计量没有检测到故障,此时模态间差异仍然存在,控制限仍然由稀疏模态确定,因而故障检测率较低。

图 4(c)使用了 KNN 方法,可以有效处理疏密程度相似的多模态数据,但在处理疏密程度差异较大的多模态数据时,控制限依然由稀疏模态确定。由图 4(c)可以看出,虽然  $D^2$  成功检测出 4 个故障,但位于密集模态的微弱故障 f4 没有被检测出来。图 4(d)使用 MLNS-NPE 方法,两个模态变为一个模态,消除模态间结构信息, $T^2$ 、 $SPE$  均成功检测出所有故障。四种方法故障检测结果如表 2 所示。

表 2 99% 控制限下四种算法对数值例子的检测结果

算法	统计量	1	2	3	4	5	故障检测率/%
NPE	$T^2$	√	×	×	×	×	20
	$SPE$	×	×	×	×	√	20
KNPE	$T^2$	√	×	×	×	√	40
	$SPE$	×	×	×	×	×	0
KNN	$D^2$	√	√	√	×	√	80
MLNS-NPE	$T^2$	√	√	√	√	√	100
	$SPE$	√	√	√	√	√	100

### 3.2 青霉素发酵过程

青霉素发酵生产过程是典型的多阶段过程,整个生产过程分为三个阶段:1) 菌体生长阶段:发酵培养基接种后青霉菌在合适的环境中经过短时间的适应开

始发育、生长和繁殖,伴随葡萄糖底料的快速消耗;2) 青霉素合成阶段:这个阶段主要合成青霉素,青霉素的生产率达到最大;3) 菌体自溶阶段:这个阶段菌体衰老,细胞开始自溶,青霉素合成能力衰退。图 5 为青霉素发酵过程工艺流程。

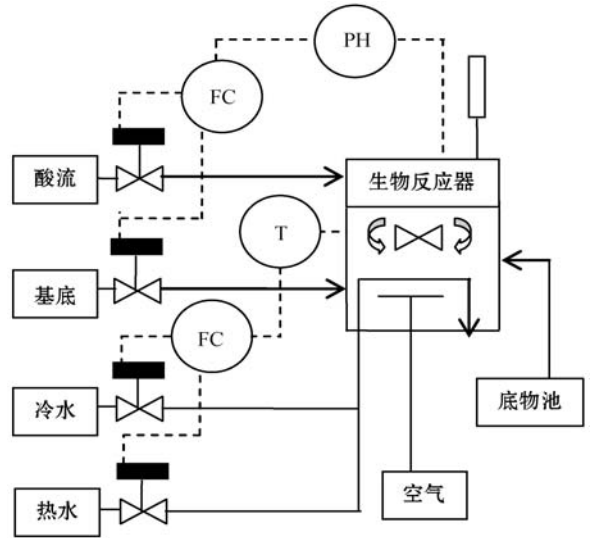


图 5 青霉素发酵工艺流程

本文通过 Pensim 仿真平台<sup>[16-17]</sup>实现青霉素发酵过程的仿真。Pensim 仿真平台可以模仿青霉素发酵生产过程,它有 5 个输入变量用来控制发酵过程中各种参数的变化,还有 9 个过程变量,产生于菌体合成以及生长过程中,另外还有影响青霉素产量的 5 个质量变量。整个生产过程分为两个大的阶段,前 43 h 为青霉素菌生长阶段,为稀疏模态数据阶段;43 ~ 400 h 为合成青霉素阶段,为密集模态数据阶段。可在前三个变量(空气流量、搅拌功率、底物流速率)上引入阶跃和斜坡故障,并且还可以设置两种故障的幅度、引入和结束时间<sup>[18-19]</sup>。

选取青霉素发酵过程的 12 个主要变量作为故障检测变量,如表 3 所示。对前三个变量施加的故障类型和幅度如表 4 所示,其中 f1 为稀疏模态故障,f2 - f6 为密集模态故障。实验时,采取仿真时间为 400 h,采样时间为 1 h,各输入参数全部采取 Pensim 平台的默认值运行一次,产生一个批次(400 × 18)的正常数据作为训练数据进行建模。

表 3 青霉素变量类型

序号	变量
1	空气流量
2	搅拌功率
3	底物流速率
4	底物流加温
5	底物浓度

续表 3

序号	变量
6	溶解氧浓度
7	CO <sub>2</sub> 浓度
8	pH 值
9	反应器温度
10	热量
11	菌体浓度
12	青霉素浓度

表 4 故障类型

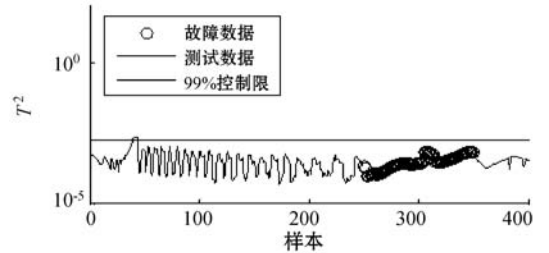
故障	变量	类型	幅度	引入时刻/h	结束时刻/h
1	1	阶跃	-0.5%	10	40
2	1	斜坡	0.3 L/h	300	400
3	2	阶跃	10%	150	300
4	2	斜坡	-5 L/h	250	350
5	3	阶跃	3%	250	400
6	3	斜坡	-0.01 L/h	300	400

应用 NPE、KNPE、KNN、MNPE 四种算法依次进行仿真,NPE、KNPE、MNPE 的重构近邻数为 6,KNN 的近邻数  $k = 13$ ,MNPE 方法第一近邻的  $k$  近邻数为 13。四种算法的故障检测率如表 5 所示。由于过程分为两个模态,且密度显著不同,不服从多元高斯分布,所以 NPE、KNPE 方法检测效率都较低。f3 为大尺度故障,因此四种算法的故障检测效率都很高。KNN 方法能够对一般多模态数据进行检测,但当数据模态差异明显时,控制限由稀疏模态确定,因此故障检测能力较低,f2、f4、f5、f6 为这类故障。f1 的类型为稀疏模态的微弱故障,其到正常样本的距离小于正常样本之间距离,因此故障检测效率比较低。使用 MNPE 算法,两个模态融合成单一模态,消除了多模态特征,与其他三种算法相比,6 组实验 MLNS-NPE 故障检测效率最高。

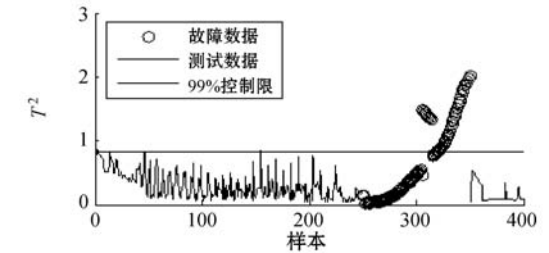
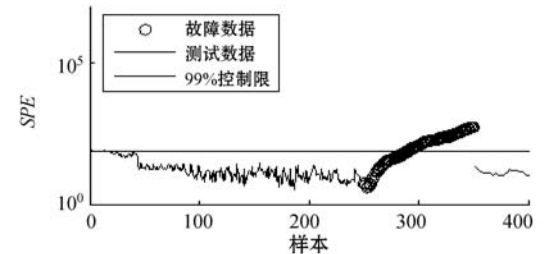
表 5 故障检测率 %

故障	NPE		KNPE		KNN	MNPE	
	$T^2$	$SPE$	$T^2$	$SPE$	$D^2$	$T^2$	$SPE$
1	3.23	51.61	9.68	38.71	51.61	96.77	100.00
2	0	59.41	0	49.50	67.33	94.06	95.05
3	0	100.00	27.15	96.69	100.00	99.34	100.00
4	0	59.41	39.60	51.49	78.22	97.03	96.04
5	0	0	0	0	0	96.69	94.70
6	6.93	0	0	0	0	94.06	92.08

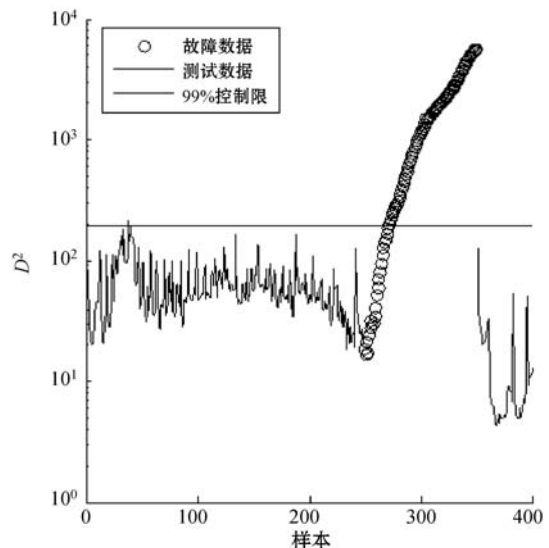
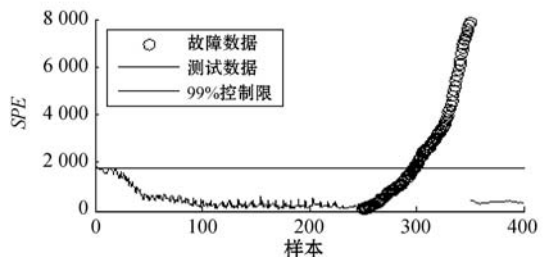
时,故障样本与正常样本间差异很小,存在故障延迟现象,因此四种算法在故障引入初始时刻的故障检测能力很弱,但所提出的 MNPE 方法的故障延迟最小,检测效率也最高。



(a) NPE 算法

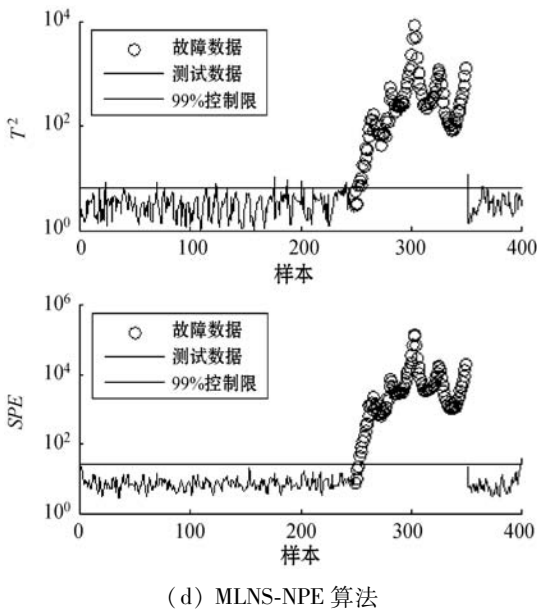


(b) KNPE 算法



(c) KNN 算法

图 6 中 f4 的故障类型为斜坡故障,故障刚引入



(d) MLNS-NPE 算法  
图6 四种算法对故障4的检测结果

## 4 结 语

本文提出一种基于改进LNS和邻域保持嵌入的算法来提高NPE算法对方差差异明显的多模态数据的故障检测能力。MLNS方法可以使疏密程度不同的多模态数据变为单一模态,并近似服从多元高斯分布,从根本上消除多模态数据影响,为后续NPE算法的使用提供先决条件。利用NPE算法降维和特征提取,同时将高维局部信息保留。将二者结合应用在数值例子和青霉素生产过程中。与其他几种算法相比,所提出的MLNS-NPE算法故障检测率最高,验证了该方法的可行性。

## 参 考 文 献

[1] Wise B M, Gallagher N B, Butler S W, et al. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process[J]. *Journal of Chemometrics*, 1999, 13(3-4): 379-396.

[2] Wilson D J H, Irwin G W. PLS modelling and fault detection on the Tennessee Eastman benchmark[C]//*Proceedings of the 1999 American Control Conference*, 1999.

[3] Roweis S T, Saul L K. Nonlinear dimensionality reduction by locally linear embedding[J]. *Science*, 2000, 290(5500): 2323-2326.

[4] Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation[J]. *Neural Computation*, 2003, 15(6): 1373-1396.

[5] 周东华,李钢,李元. 数据驱动的工业过程故障诊断技术[M]. 北京:科学出版社,2011.

[6] Bengio Y, Paiement J F, Vincent P, et al. Out-of sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering[C]//*16th International Conference on Neural Information Processing Systems*, 2003.

[7] He X, Cai D, Yan S, et al. Neighborhood preserving embedding[C]//*Tenth IEEE International Conference on Computer Vision*, 2005.

[8] Liu H, Lai Z, Chen Y. Joint sparse neighborhood preserving embedding[J]. *Journal of Physics: Conference Series*, 2019, 1176:032023.

[9] 赵小强,王涛. 基于TGNPE算法的间歇过程故障诊断[J]. *化工学报*, 2016, 67(3): 1055-1062.

[10] 杨健,宋冰,谭帅,等. 时序约束NPE算法在化工过程故障检测中的应用[J]. *化工学报*, 2016, 67(12): 5131-5139.

[11] 寇勃晨,唐力伟,邓士杰. 改进模型的自适应NPE算法故障降维辨识[J]. *兵器装备工程学报*, 2018, 39(8): 97-102.

[12] 宋冰,马玉鑫,方永锋,等. 基于LSNPE算法的化工过程故障检测[J]. *化工学报*, 2014, 65(2): 620-627.

[13] Ma H, Hu Y, Shi H. A novel local neighborhood standardization strategy and its application in fault detection of multi-mode processes[J]. *Chemometrics and Intelligent Laboratory Systems*, 2012, 118(7): 287-300.

[14] 冯立伟,张成,李元,等. 基于双近邻标准化和PCA的多阶段过程故障检测[J]. *化工学报*, 2018, 69(7): 3159-3166.

[15] Xue Y, Wang Y, Yang J. An independent component analysis algorithm through solving gradient equation combined with kernel density estimation[J]. *Journal of Shanghai Jiaotong University (Science)*, 2009, 14(2): 204-209.

[16] Lee J M, Yoo C K, Lee I B. Fault detection of batch processes using multiway kernel principal component analysis[J]. *Computers & Chemical Engineering*, 2004, 28(9): 1837-1847.

[17] Tan S, Wang F, Peng J, et al. Multimode process monitoring based on mode identification[J]. *Industrial & Engineering Chemistry Research*, 2011, 51(1): 374-388.

[18] 刘毅,王海清. Pensim仿真平台在青霉素发酵过程的应用研究[J]. *系统仿真学报*, 2006, 18(12): 3524-3527.

[19] Wang G, Liu J, Li Y, et al. Fault detection based on diffusion maps and k nearest neighbor diffusion distance of feature space[J]. *Journal of Chemical Engineering of Japan*, 2015, 48(9): 756-765.