一种改进的局部切空间排列算法

顾艳春

(佛山科学技术学院 电子与信息工程学院, 广东 佛山 528000)

摘　要: 局部切空间排列(LTSA)算法是一种有效的流形学习算法,能较好地学习出高维数据的低维嵌入坐标。数据点的切空间在 LTSA 算法中起着重要的作用,其局部几何特征多是在样本点的切空间内表示。但是在实际中,LTSA 算法是把数据点邻域的样本协方差矩阵的主元所张成的空间当做数据点的切空间,导致了在非均匀采样或样本邻域均值点与样本自身偏离程度较大时,原算法的误差增大,甚至失效。为此,提出一种更严谨的数据点切空间的计算方法,即数据点的邻域矩阵按照数据点本身进行中心化。通过数学推导,证明了在一阶泰勒展开的近似下,提出的计算方法所得到的空间即为数据点自身的切空间。在此基础上,提出了一种改进的局部切空间排列算法,并通过实验结果体现了该方法的有效性和稳定性。与已有经典算法相比,提出的计算方法没有增加任何计算复杂度。

关键词: 流形学习; 数据降维; 局部切空间排列; 切空间; 协方差矩阵

中图分类号: TP391.4　　**文献标志码**: A　　**文章编号**: 1001-3695(2013)03-0728-04

doi:10.3969/j.issn.1001-3695.2013.03.021

Improved local tangent space alignment algorithm

GU Yan-chun

(*School of Electronics & Information Engineering*, *Foshan University*, *Foshan Guangdong* 528000, *China*)

Abstract: As one of the classical manifold learning algorithms, LTSA algorithm can yield low-dimensional embedding coordinates from high-dimensional space effectively. Tangent space plays a central role in LTSA algorithm by projecting each neighborhood into the tangent space to obtain the local coordinates. However, in practice, LTSA algorithm takes the space which spanned by principal components of the sample covariance matrix of the neighborhood as the tangent space of the point. This paper presented a more rigorous method to calculate tangent space, that the neighborhood matrix of data points was centralized in accordance with the data point itself. By mathematical deduction, it proved that, under the approximation of first order Taylor, the space attained by our method is even the tangent space of data points itself. Based on this method, it proposed an improved local tangent space alignment algorithm. The effectiveness and stability of this algorithm are further confirmed by some experiments. Moreover, the proposed algorithm has no increase in the computational complexity.

Key words: manifold learning; data reduction; local tangent space alignment(LTSA); tangent spaces; covariance matrix

0　引言

随着计算机和网络的广泛使用,人们可以方便地获取大量的信息,随之出现了表示信息的数据集的维数越来越高、非结构化更突出等特点[1]。对于这种高维数据集,人们无法直接感知其内在规律,需要借助数据分析和降维方法来帮助人们理解数据。流形学习就是利用数据集的局部几何特性来揭示内在的流形结构,以达到数据降维的目的,因此,受到了机器学习和认知科学领域研究者的广泛关注[2~4]。

自 2000 年《Science》上刊出了使用 isometric feature mapping(ISOMap)[5] 和 locally linear embedding(LLE)[6] 等方法进行流形学习的算法与实验之后,流形学习真正成为研究者关注的热点。在流形学习的众多方法中,嵌入方法是其中一个热点。嵌入方法的基本思想是将流形的局部与全局特征量化成样本之间的数量关系,然后试图在低维空间中寻找嵌入向量集,使得这种低维关系在高维空间中尽可能保持下来。目前出现的嵌入方法主要有 ISOMap、LLE、Hessian Eigenmaps(HLLE)[7]、LTSA[8]、Laplacian eigenmaps(LE)[9]、Diffusion Maps[10] 等。这其中,局部切空间排列(LTSA)算法是一种很好的嵌入流形学习算法,能较好地学习到高维数据集的低维坐标。其基本思想是利用样本点邻域的切空间来表示局部几何性质,观测数据点在局部切空间的投影,获得局部低维坐标,然后将这些局部切空间排列起来构造流形的全局低维坐标[11]。

随后,在 LTSA 算法的基础上,陆续出现了一些改进的 LTSA 算法。针对 LTSA 算法受流形曲率影响较大的问题,Zhang 等人[12] 将流形的局部曲率引入 LTSA 的极小化模型中,提出了自适应 LTSA 算法,减小了 LTSA 在流形曲率较大时构造全局嵌入坐标的偏差。针对 LTSA 处理大容量样本集效率较低的问题,Yang 和 Dong 等人[13,14] 分别提出了基于划分的 LTSA 算法和基于最大线性片划分的 LTSA 算法,分别利用改进的 K-均值聚类算法和从局部线性切空间逼近的角度将高维数据划分为彼此交叠的块,计算每一块的局部切空间坐标,然后利用旋转、缩放、平移等变换排列成全局低维坐标。针对 LTSA 算法无法有效处理新来样本点的问题,Zhang 等人[15] 提出了线性的局部切空间排列算法。在所有这些算法中,数据点的切空间扮演着重要的作用,即样本点的嵌入向量集多是在数据点的切空间里表示。然而在实践中,上述算法的数据点切空间的计算方法均是把数据点邻域的样本协方差矩阵的主元当做数据点

收稿日期: 2012-07-06; **修回日期**: 2012-08-10

作者简介: 顾艳春(1981-),男,河南信阳人,博士,主要研究方向为机器学习及多媒体技术(gufenger@fosu.edu.cn).

切空间的标准正交基。本文提供的数学证明表明,通过上述方法计算的空间,不是数据点的切空间,充其量只是数据点邻域的均值点的切空间。在一些非均匀采样的流形学习问题中,数据点邻域的均值点有可能远远偏离数据点,这时其计算方法就会造成较大的误差,甚至失效。

针对此问题,本文提出一种更严谨的数据点切空间的计算方法,并在此基础上改进了 LTSA 算法。在本文提出的切空间计算方法中,数据点的邻域矩阵按照数据点本身进行中心化,并以此中心化矩阵的主元作为样本点切空间的标准正交基。通过数学推导,本文证明,在一阶泰勒展开的近似下,根据本文提出的计算方法所得到的空间为数据点自身的切空间。以此计算方法为基础,本文提出了改进的 LTSA 算法。与原 LTSA 算法相比,本文的算法更能体现样本点切空间的概念,能够减小全局重构误差。实验结果表明了改进的 LTSA 算法对非均匀采样或邻域值取值较小的样本集有更好的有效性和稳定性。

1　改进的局部切空间排列算法

对于非线性流形来说,全局的非线性结构来自于局部线性分析和局部线性信息的全局整合[16],根据这一思想,2004 年张振跃等人提出 LTSA 算法,该算法通过逼近每一个样本点的切空间来构建低维流形的局部几何,观测数据点在局部切空间的投影来获得局部坐标,然后将这些局部坐标排列起来构造流形的全局坐标。切空间在 LTSA 算法中扮演着非常重要的作用。计算切空间时,LTSA 算法采取邻域内各点减去邻域均值点后的中心化矩阵来获取切空间坐标。本文提供的数学证明表明,这样计算的空间,不是数据点的切空间,充其量只是数据点邻域均值点的切空间。在一些非均匀采样的流形学习问题中,数据点邻域的均值点有可能远远偏离数据点,其数据点切空间的计算方法就会造成较大的误差,甚至失效。本文用样本点自身取代中心化矩阵中的均值点来获取切空间坐标。通过推导,本文证明,在一阶泰勒展开的近似下,根据本文提出的计算方法所得到的空间就是数据点的切空间,在此基础上提出了改进的 LTSA 算法(简称 ILTSA)。

1.1　切空间计算方法

设 $M\subseteq R^D$ 表示一个黎曼流形,$\Xi\subseteq R^d$ 表示一个开域,这里 $d<<D$。又设 $\varphi:\Xi\to M$ 是一个拓扑同胚映射。给定 $\tilde{x}\in M$,记 $\tilde{y}\in\Xi$,使得 $\varphi(\tilde{y})=\tilde{x}$。下面说明怎样计算 $\tilde{x}$ 点切空间 $T_{\tilde{x}}(M)$ 的标准正交基。可以证明,对于任意的 $\tilde{x}\in M$,记 $\tilde{y}\in\Xi$,使得 $\varphi(\tilde{y})=\tilde{x}$,则 φ 在 $\tilde{y}$ 点的 Jacobi 矩阵 $J_\varphi(\tilde{y})$ 的列向量构成 $\tilde{x}$ 点的切空间 $T_{\tilde{x}}(M)$ 的基。虽然 Jocobi 矩阵 $J_\varphi(\tilde{y})$ 的列向量构成了 $\tilde{x}$ 点的切空间 $T_{\tilde{x}}(M)$ 的基,但是因为 φ 未知,因此 Jocobi 矩阵 $J_\varphi(\tilde{y})$ 也未知,需要通过另外的方法求 $T_{\tilde{x}}(M)$。

设 $x_k\in M$ 是 $\tilde{x}$ 点的一个邻域,又设 $y_k\in\Xi$,使得 $\varphi(y_k)=x_k$,其中 $k=1,\cdots,K,d<<K$。记 $\tilde{X}=[x_1-\tilde{x}\quad\cdots\quad x_K-\tilde{x}]$,$\tilde{Y}=[y_1-\tilde{y}\quad\cdots\quad y_K-\tilde{y}]$。

在一阶泰勒展开的近似下,有

$$x_k-\tilde{x}=\varphi(y_k)-\varphi(\tilde{y})\approx J_\varphi(\tilde{y})(y_k-\tilde{y})\quad k=1,\cdots,K$$

写成矩阵的形式,有

$$\tilde{X}=[x_1-\tilde{x}\quad\cdots\quad x_K-\tilde{x}]\approx J_\varphi(\tilde{y})[y_1-\tilde{y}\quad\cdots\quad y_K-\tilde{y}]=J_\varphi(\tilde{y})\tilde{Y}$$

另一方面,对 $\tilde{X}$ 进行 SVD 分解,有 $\tilde{X}=U\Sigma V^{\mathrm{T}}$。其中:$U$ 和 V 都是列向量标准正交的矩阵,U 的大小为 $D\times r$,V 的大小为 $K\times r$,Σ 为 $r\times r$ 的对角矩阵,对角线上的元素大于零,r 是矩阵 $\tilde{X}$ 的秩。

于是,利用 $\tilde{X}=U\Sigma V^{\mathrm{T}}\approx J_\varphi(\tilde{y})\tilde{Y}$,有

$$U\approx J_\varphi(\tilde{y})\tilde{Y}V\Sigma^{-1}=J_\varphi(\tilde{y})\Theta$$

其中:$\Theta=\tilde{Y}V\Sigma^{-1}$。上式表明,$U$ 的列向量是 $J_\varphi(\tilde{y})$ 列向量的线性组合,因此,U 的列向量张成的空间是 $J_\varphi(\tilde{y})$ 的列向量张成的空间的子空间,也就是 $\tilde{x}$ 点的切空间 $T_{\tilde{x}}(M)$ 的一个子空间。如果 $r=d$,也就是邻域矩阵 $\tilde{X}$ 的秩为 d,则 U 的列向量就构成 $\tilde{x}$ 点切空间 $T_{\tilde{x}}(M)$ 的标准正交基。

1.2　切空间计算方法比较

在数据点切空间的计算方法上,LTSA 算法是基于样本点邻域的均值点来求切空间,本文的方法是利用样本点本身来求切空间。记 $\mu_{\tilde{x}}=\frac{1}{K}\sum_{k=1}^{K}x_k$,$\tilde{X}=[x_1-\tilde{x}\quad\cdots\quad x_K-\tilde{x}]$,$\hat{X}=[x_1-\mu_{\tilde{x}}\quad\cdots\quad x_K-\mu_{\tilde{x}}]$。

$\tilde{X}$ 与 $\hat{X}$ 的区别在于:$\tilde{X}$ 中的邻域以数据点 $\tilde{x}$ 本身进行中心化,而 $\hat{X}$ 中的邻域则以数据点邻域的均值 $\mu_{\tilde{x}}$ 进行中心化。本文提出的方法是对 $\tilde{X}$ 进行 SVD 分解,而 LTSA 算法则是对 $\hat{X}$ 进行 SVD 分解。理论上,本文提出的方法得到 $\tilde{x}$ 点的切空间 $T_{\tilde{x}}(M)$ 的标准正交基,而 LTSA 算法求取切空间的方法在 $\mu_{\tilde{x}}=\frac{1}{K}\sum_{k=1}^{K}x_k\in M$、$\mu_{\tilde{y}}=\frac{1}{K}\sum_{k=1}^{K}y_k\in\Xi$ 和拓扑同胚映射 φ 的同时还是线性映射的三个假设下,得到均值点 $\mu_{\tilde{x}}$ 的切空间 $T_{\mu_{\tilde{x}}}(M)$ 的标准正交基。事实上,在上述的三个假设下,有

$$\varphi(\mu_{\tilde{y}})=\varphi\left(\frac{1}{K}\sum_{k=1}^{K}y_k\right)=\frac{1}{K}\sum_{k=1}^{K}\varphi(y_k)=\frac{1}{K}\sum_{k=1}^{K}x_k=\mu_{\tilde{x}}$$

进而有

$$\begin{aligned}\hat{X}&=[x_1-\mu_{\tilde{x}}\quad\cdots\quad x_K-\mu_{\tilde{x}}]=\\&[\varphi(y_1)-\varphi(\mu_{\tilde{y}})\quad\cdots\quad\varphi(y_K)-\varphi(\mu_{\tilde{y}})]\approx\\&J_\varphi(\mu_{\tilde{y}})[y_1-\mu_{\tilde{y}}\quad\cdots\quad y_K-\mu_{\tilde{y}}]=J_\varphi(\mu_{\tilde{y}})\hat{Y}\end{aligned}$$

其中:$\hat{Y}=[y_1-\mu_{\tilde{y}}\quad\cdots\quad y_K-\mu_{\tilde{y}}]$。

另一方面,对 $\hat{X}$ 进行 SVD 分解,有

$$\hat{X}=Q\Lambda P^{\mathrm{T}}$$

进而有

$$Q\approx J_\varphi(\mu_{\tilde{y}})\hat{Y}P\Lambda^{-1}$$

上式表明,对 $\hat{X}$ 进行 SVD 分解得到的空间其实是均值点 $\mu_{\tilde{x}}$ 的切空间 $T_{\mu_{\tilde{x}}}(M)$,而不是 $\tilde{x}$ 点的切空间 $T_{\tilde{x}}(M)$。

1.3　改进的局部切空间排列算法

针对原 LTSA 算法在切空间计算方法上的不足,本文给出了一种改进的 LTSA 算法(ILTSA)。

给定高维空间的 N 个数据点 $X=[x_1,x_2,\cdots,x_N]$,$x_i\in\mathrm{R}^m$,ILTSA 算法的第一步也是寻找每个样本点的邻域,但其邻域不包括原样本点。设 $X_i=[x_{i_1},\cdots,x_{i_k}]$ 为不包括样本点 x_i 自身在内的最近 k 个邻域点构成的矩阵。ILTSA 算法的第二步是计算邻域内各点在 x_i 点的切空间的投影来逼近邻域内各点,从而获得局部坐标,即

$$\min_{x,\Theta,Q}\sum_{j=1}^{k}\|x_{ij}-(x+Q\theta_j)\|_F^2=\min_{x,\Theta,Q}\sum_{j=1}^{k}\|X_i-(xe^{\mathrm{T}}+Q\Theta)\|_F^2\tag{1}$$

其中:x 由 x_i 给出,即样本自身点;Q 可由 Q_i 给出;Q_i 为 $X_i-x_ie^{\mathrm{T}}$ 的 d 个最大的奇异值对应的左奇异向量。Θ 由 Θ_i 给出:

$$\Theta_i=Q_i^{\mathrm{T}}(X_i-x_ie^{\mathrm{T}})=[\theta_1^{(i)},\cdots,\theta_k^{(i)}]\tag{2}$$

其中:$\theta_j^{(i)}=Q_i^{\mathrm{T}}(x_{i_j}-x_i)$。

ILTSA 的第三步是将这些交叠的局部坐标系统排列起来

得到一个全局坐标系统。类似原 LTSA 算法的构造方式,可得到 ILTSA 算法的排列矩阵为

$$B = HH^{\mathrm{T}} \tag{3}$$

其中:$H_i = Z_i(I - (\Theta_i)^{+}\Theta_i)$,$Z_i = S_i - T_i\Gamma^{\mathrm{T}}$,$S_i$和 T_i 为 0-1 选择矩阵。S_i 的大小为 $N \times k$,每一列中,除第 i 个元素为 1 外,其他元素均为 0;T_i 的大小为 $N \times 1$,除第 i 个元素为 1 外,其他元素均为 0。这样,极小化重建误差的最优解能通过计算矩阵 B 的第 $2 \sim d+1$ 个最小特征值对应的特征向量 $u_2, \cdots, u_{d+1}$ 来获得,即 $T = [u_2, \cdots, u_{d+1}]^{\mathrm{T}}$。

改进的局部切空间排列算法(ILTSA)如下:

a)选取邻域。计算每个样本点 x_i 的邻域。记 $X_i = [x_{i_1}, \cdots, x_{i_k}]$为不包括样本点自身在内的最近 k 个邻域点。

b)局部线性投影。对每个样本点的邻域,计算以样本点自身中心化矩阵的最大 d 个奇异值对应的右奇异向量,并将这 d 个右奇异向量组成矩阵 V_i。

c)局部坐标排列成全局坐标。按式(3)构造排列矩阵 $B = HH^{\mathrm{T}}$。计算 B 的第 $2 \sim d+1$ 个最小特征值对应的特征向量,则为计算的嵌入结果。

2 实验分析

为了更好地比较和分析 ILTSA 与原 LTSA 算法的效果及性能差异,本文设计了以下的实验。实验中,CPU 频率为 1.86 GHz,内存容量为 2 GB,运行环境为 MATLAB 7.0。

2.1 运行效果及分析

首先选取 Mani 程序中的标准数据集。Mani 数据集是一种在流形学习中广泛使用的数据集,可以方便地从 http://www.math.ucla.edu/~wittman/mani/index.html 处免费下载。当样本点 N 取 800、邻域值 K 取 8 时,对比效果如图 1 所示。由图 1 可以看出,采用样本点本身中心化的算法,其效果与原 LTSA 算法效果相比基本类似。这主要是因为样本点较多,样本点与邻域点的距离较小,而邻域值取值较大,使得数据点邻域的均值点与数据点本身差距不大。而对于 Punctured Sphere 这类非均匀采样的样本集,ILTSA 算法取得了更好的效果,更好地表现了流形的本质特征。

当样本点取 400、邻域值取 4 时,对比效果如图 2 所示。由图 2 可以看出,原 LTSA 算法基本失效。这主要是因为样本点较少,样本点与其邻域点的距离较远,此时,减小邻域值为 4 时,邻域内均值点就与样本点偏差较大,求出的样本点的低维坐标不能准确地反映样本点与邻域点的真实关系。采用样本点本身中心化的算法,其效果与原 LTSA 算法效果相比有较大进步,显得更稳定和有效。这主要是因为本文的算法思想是在样本点的切空间内,尽可能地用邻域内各点来逼近数据点本身,而不是均值点。对于 S-Curve 数据集和其他流形数据集,实验结果较为类似。当样本点较多且邻域值较大时,ILTSA 算法与原算法的效果相近;当样本点较少且邻域值较小时,ILTSA 算法表现了较好的效果,这表明了改进的 LTSA 算法的健壮性和稳定性。实验效果分别如图 3 和 4 所示。

同时,本文的改进算法也能应用在真实数据集上,如图 5 所示。本文选取了文献[5]中的人脸数据集。由图 5 可知,改进的 LTSA 算法能较好地寻找到高维流形的本质低维特征。

2.2 算法效率

在比较和分析两种算法效果差异的同时,本文作了算法效率的对比分析,如表 1 所示。理论上可以证明,本文的算法与原算法具有相同的时间复杂度,实验结果证实了这一方法的有效性。

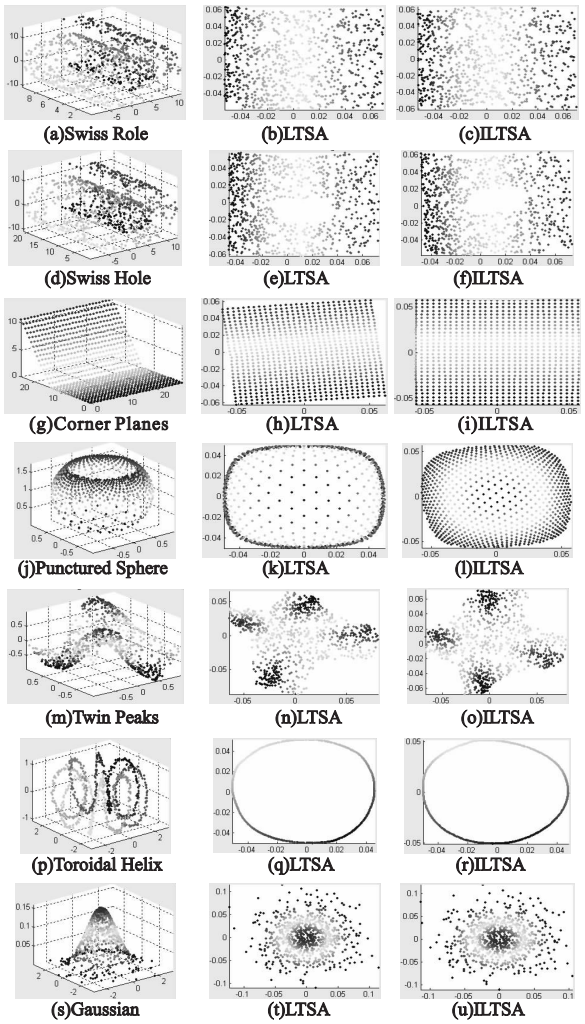

图1 Mani数据集算法效果对比(N=800,K=8)

表 1 算法执行时间对比表($N = 800, K = 8$)

流形数据集	原 LTSA 算法/s	ILTSA 算法/s
Swiss Role	0.703	0.687
Swiss Hole	0.703	0.688
Corner Planes	0.844	0.813
Punctured Sphere	0.719	0.75
Twin Peaks	0.687	0.672
Toroidal Helix	0.703	0.687
Gaussian	0.687	0.671

由表 1 可以看出,与原 LTSA 算法相比,样本点本身取代邻域均值进行中心化的切空间计算方法没有增加计算负担。

3 结束语

LTSA 算法是一种非常经典的流形学习算法,能较好地学习到高维数据流形的低维坐标。理论上,LTSA 算法基于数据点的切空间,但是在实践中,LTSA 算法采用数据点邻域的样本协方差矩阵的主元所张成的空间作为数据点的切空间。本文针对 LTSA 算法中对于样本点切空间计算方法的隐患,提出了一种更为严谨的计算方法,并在此基础上改进了 LTSA 算法。实验结果证实了改进算法的有效性和稳定性,而且本文提出的数据点切空间的计算方法没有增加计算负担。同时,本文提到的样本切空间的计算方法也可应用到其他以切空间为基础的流形学习算法(如 HLLE 算法)中。与其他流形学习算法一样,LTSA 算法不能很好地解决新来样本点的问题,本文提出的改进算法也未能处理这一问题,这将是笔者未来的研究工作之一。另外,建立样本点本身与邻域均值点的偏离程度对于全局坐标影响的数值分析和模型,也将是笔者未来的重要工作。

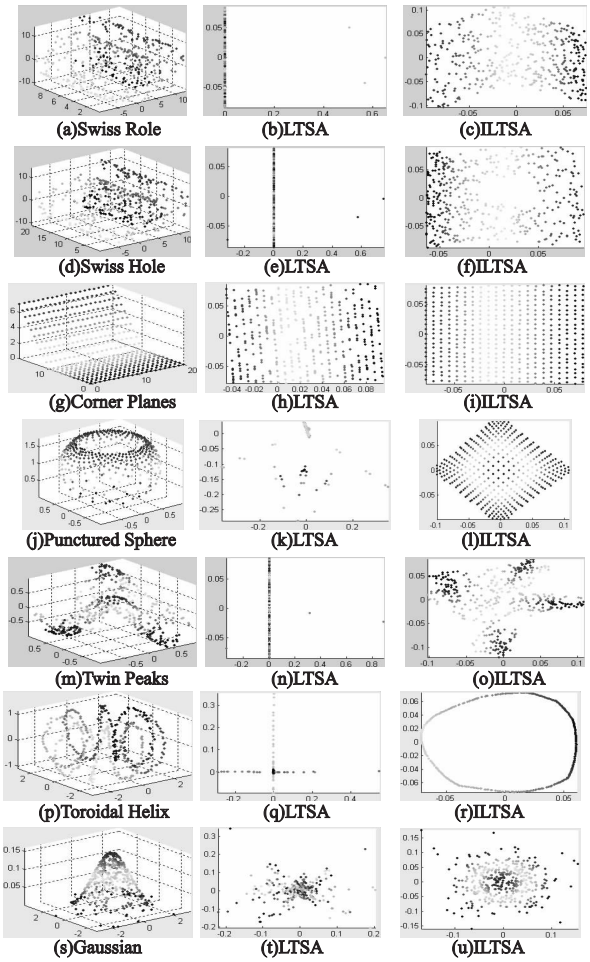

图2　Mani数据集算法效果对比（N=400，K=4）

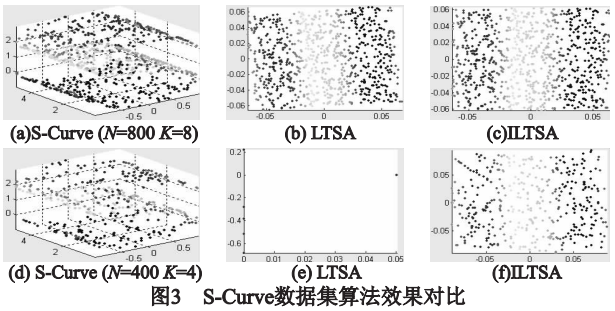

图3　S-Curve数据集算法效果对比

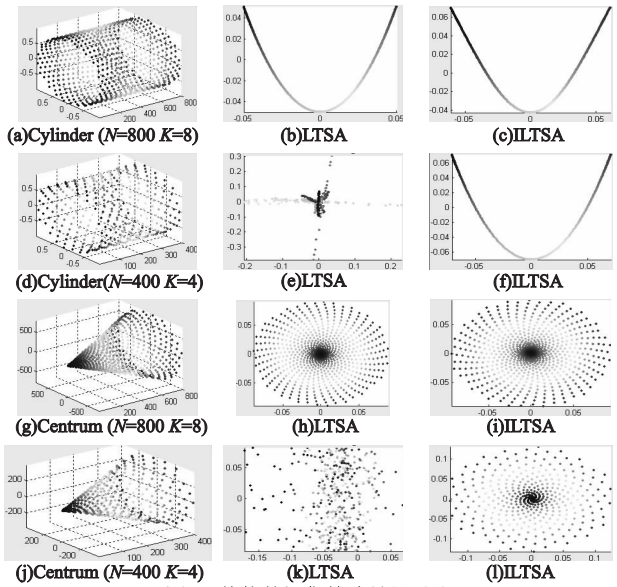

图4　其他数据集算法效果对比

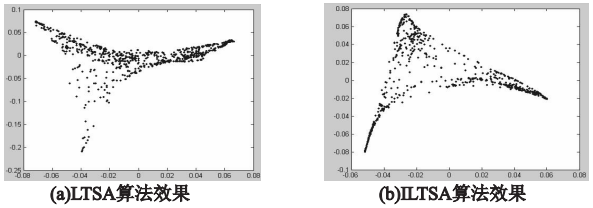

图5　真实数据集算法效果对比

参考文献:

[1] SEUNG H S, LEE D D. Cognition: the manifold way of perception [J]. Science, 2000, 290(5000): 2268-2269.

[2] ZHANG Tian-hao, TAO Da-cheng, LI Xue-long, *et al*. A unifying framework for spectral analysis based dimensionality reduction[C]// Proc of IEEE International Joint Conference on Neural Networks. 2008: 1670-1677.

[3] Van der MAATEN L J P, POSTMA E O, Van den HERIK H J. Dimensionality reduction: a comparative review[J]. Pattern Recognition, 2007, 10: 1-41.

[4] DONOHO D L. High dimensional data analysis: the curse and blessings of dimensionality[C]//Proc of American Math Society on Math Challenges of the 21st Century. 2000.

[5] TENENBAUM J B, SILVA V D, LANGFORD J C. A global geometric framework for nonlinear dimensionality reduction[J]. Science, 2000, 290(5000): 2219-2323.

[6] ROWEIS S T, SAUL L K. Nonlinear dimensionality reduction by locally linear embedding [J]. Science, 2000, 290 (5000): 2323-2326.

[7] DONOHO D L, GRIMES C. Hessian eigenmaps: locally linear embedding techniques for high-dimensional data[J]. Proceedings of the National Academy of Sciences, 2003, 100(10): 5591-5599.

[8] ZHANG Z Y, ZHA H Y. Principal manifolds and nonlinear dimension reduction via local tangent space alignment[J]. SLAM Journal of Scientific Computing, 2004, 26(1): 313-338.

[9] BELKIN M, NIYOGI P. Laplacian eigenmaps for dimentionality reduction and data representation[J]. Neural Computation, 2003, 15(6): 1373-1396.

[10] COIFMAN R R, LAFON S. Diffusion maps[J]. Applied and Computational Harmonic Analysis, 2006, 21(1): 5-30.

[11] LIU Xiao-ming, YIN Jian-wei, FENG Zhi-lin, *et al*. Incremental manifold learning via tangent space alignment[C]//Proc of ANNPR. 2006: 107-121.

[12] ZHANG Zhen-yue, WANG Jing, ZHA Hong-yuan. Adaptive manifold learning[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2012, 34(2): 253-265.

[13] YANG Jian, LI Fu-xin, WANG Jue. A better scaled local tangent space alignment algorithm[C]//Proc of IEEE International Joint Conference on Neural Networks. 2005: 1006-1011.

[14] DONG Ji-yuan, MU Zhi-chun, OUYANG D H. An imporved local tangent space alignment algorithm based on max linear patch partition and its application in multi-pose ear recognition[C]//Proc of the 29th Chinese Control Conference. 2010: 3062-3067.

[15] ZHANG Tia-hao, YANG Jie, ZHAO De-li. Linear local tangent space alignment and application to face recognition[J]. Neurocomputing, 2007, 70(79): 1547-1553.

[16] ZHA Hong-yuan, ZHANG Zhe-yue. Spectral properties of the alignment matrices in manifold learning[J]. SIAM Review, 2009, 51(3): 545-566.