自组织拓扑映射与主曲线学习*)

倪劲松[1] 李玉珍[2] 王宜怀[2]
(苏州大学数学科学学院[1] 计算机科学与技术学院[2] 苏州 215006)

摘 要 本文利用自组织拓扑映射方法设计了一种简易主曲线学习的算法,该算法继承了HS主曲线算法和K主曲线算法的主要优点,同时降低了一般主曲线算法的难度,使其变得更简洁明了。

关键词 向量量化器,自组织拓扑映射,Voronoi邻域,主曲线

Self-organized Topological Mapping and Principal Curve Learning

NI Jin-Song LI Yu-Zheng WANG Yi-Huai
(Mathematics and Information Science College, Suzhou University, Suzhou 215006)

Abstract We use the method of self-organized topological mapping to design a learning algorithm of principal curves. The algorithm is composed by two parts. In the first part, based on the theory of generalized vector guantizer, we have achieved refined GL-algorithm, and use it to compute polygonal line principal curves for every finite discrete data set, which is like Kegl has done. In the second part, we combine the method of self-organized topological mapping with refined GL-algorithm to design an algorithm of principal curves(Algorithm C), which is simpler than HS-type and K-type. This algorithm has inherited main virtues of the principal curves of HS-type and K-type.

Keywords Vector quantizer, Self-organized topological mapping, Voronoi neighborhood, Principle curve

作为线性主成分分析(PCA)的非线性推广,主曲线概念由T. Hastie和W. Stuetzle经过长达5年的研究后于1989年率先给出[7]。之后,基于其广泛的应用前景,世界各地的科学家对主曲线的理论和算法进行了一系列的改进和拓展。1992年,Banfield和Raftery给出的BR主曲线[1],虽然对HS主曲线做了改进,但误差较大。随后,产生了T主曲线[9]、D主曲线[4,5]。1999年,B. Kegl等人提出多角形算法[8],使主曲线理论跨入了一个新的里程碑。1998～2001年,Chang和Ghosh又利用自组织拓扑映射定义了概率主曲线和概率主曲面[2,3],2001～2003年,Delicado考察了线性主成分的其他特性,给出了主定向点概念和基于主定向点概念的主曲线[4,5]。

十几年的发展,来自世界各地数十位科学家发表了近百篇论文,致力于主曲线理论的研究,解决了不少HS主曲线理论中存在的一些问题,但依然处于理论发展的初期阶段。无论从理论还是应用角度都存在着较大的发展空间。在我国,主曲线理论几乎还处于学习阶段。这与国际上蓬勃发展的形势有一定的落差。本文拟在对主曲线的理论研究上作一些探索性的研究,改良了通常的GL算法,并基于此算法,利用自组织拓扑映射方法给出了较为简易的主曲线算法。

本文所引的主曲线方面基础知识可参见文[7,8],神经网络方面的基础知识可参见文[6,11]。

1 预备知识

1.1 向量量化器

定义1 一个k点向量量化器是一个映射$q:R^d\to R^d$,使得对每个输入样本$x\in R^d$,有:$q(x)\in C=\{v_1,v_2,\cdots,v_k\}\subseteq R^d$ $(k\leqslant d)$。则称C为一个k-字码书,而$v_j=q(x)(j=1,2,\cdots,k)$称为x所对应的码字。

这个k点量化器完全由$V_j=q^{-1}(v_j)(j=1,\cdots,k)$所决定的空间分类所决定。考虑$v_j=q(x)(1\leqslant j\leqslant k)$,失真度量定义为$\Delta(x,v_j)=\|x-v_j\|^2$。如将量化器$q$应用到随机变量$X\in R^d$上,则取期望失真为:

$$\Delta(q)=E(\Delta(X,q(X)))$$

向量的量化是利用输入向量的固有结构进行数据压缩的技术。具体地,输入空间被分成一些不同的区域,并且对每个区域定义一个重建向量即码字。当一个新的输入向量提供给量化器时,首先确定向量所在的区域并且利用该区域的重建向量表示输入向量。这样,使用码字编码替代原始输入向量来存储或传输,以一定的失真代价可实现在存储或传输带宽的重大节省。

一个有最小编码失真的向量量化器被称作Voronoi单元或最邻近量化器。

向量量化器的优化

定义2 如果对$\chi_n=\{x_1,x_2,\cdots,x_n\}\in R^d$上任何$k$点量化器$q$均有$\Delta(q^*)\leqslant\Delta(q)$,那么称量化器$q^*$是整体最优化。

可以证明:如随机变量X具有有限二阶矩,那么q^*存在。但一般地q^*未必唯一。通常在给定分布或密度函数的前提下,寻求一个整体最优的量化器q^*是非常困难的事。即使对于诸如Gauss,Laplace和一致分布这样最常用的模型,当$k>2$时的码书也不一定是最优向量量化器具体事例。

由于整体最优性很难达到,因此通常退而求其次去寻求局部最优的向量量化器。在此之前还是介绍一些最优量化器的相关理论。

定义3 如果$\Delta(q)$有一个极小值q^*,那么量化器q^*称为局部最优的。

在实际中,随机变量X的分布通常是不知道的,因此,我们只能设计一种称为经验量化器的码书,即寻找在离散数据集$\chi_n=\{x_1,x_2,\cdots,x_n\}\subset R^d$上的向量量化器。在此情形下,数据集$\chi_n$上$q$的经验失真度量定义为

$$\Delta_n(q)=\frac{1}{n}\sum_{j=1}^{n}\Delta(x_j,q(x_j))$$

*)基金编号:江苏省教育厅自然科学基金资助项目(02KJD52001)。倪劲松 博士,研究方向:代数拓扑、数据的几何特征挖掘。

经验量化器的设计思想是寻求一个使 $\Delta_n(q)$ 尽量小的量化器的有效算法。

理论上讲，由于数据集 χ_n 的分类是有限的，因此寻求经验的最优向量量化器是可能的。当然，给出所有不同的分类是不容易实现的，这里只是介绍一种利用迭代逼近实现寻求最优向量量化器的思想。

首先，分析

$$\Delta(q)=\Delta(\chi_n,q)=\frac{1}{n}\sum_{j=1}^{n}\Delta(x_j,\ q(x_j))$$

何时达到最小是很有意义的。如果 x_j 是具有有限二阶矩(当然 χ_n 肯定满足)，为了发现 q^*，需要在所有 k 点量化器 q 的集合 $Q(k)$ 上极小化 $\Delta(q,\chi_n)$，即

$$q_n^*=\arg\min_{q\in Q(k)}\frac{1}{n}\sum_{j=1}^{n}\|x_j-q(x_j)\|$$

其次，引入 Voronoi 区域(或集合)的概念。设 $v_i\in C$ 是码本 C 中的一个码字，定义 v_i 所对应的 Voronoi 区域为

$$V_i=\{x\in R^d\mid\Delta(x,v_i)\leqslant\Delta(x,v_j),j=1,2,\cdots,k\}$$

这样得到输入空间 R^d 的一个"划分" $\{V_1,V_2,\cdots,V_k\}$，k 为码字个数。

反之，如 R^d 有一个划分 $\{V_i\mid 1\leqslant i\leqslant k\}$，同样可以构造与此分类相对应的码书(codebook)，即

$$v_i=\arg\min_v E[\Delta(x,v)\mid x\in V_i]$$

此时也称 v_i 为 Voronoi 区域 V_i 的引力中心，它使期望失真极小化。

综上所述，如果码本 C 满足如下三个性质：

1. $V_i=\{x\in R^d\mid\Delta(x,v_i)\leqslant\Delta(x,v_j),1\leqslant j\leqslant k\}$；

2. $v_i=\arg\min_v E[\Delta(x,v)\mid x\in V_i]$；

3. $P\{x\mid x\in v_i,\Delta(x,v_i)=\Delta(x,v_j),i\neq j\}=0$(即使得处于两个相邻分类的公共边界上的数据点的概率为零)。

那么，该码本 C 被广泛地认为能够达到局部最优。当然，更一般的理论还有待于进一步的研究。

计算局部最优码书的较好方法便是下面介绍的 GL 算法：

算法 A(分布情形的 GL 算法)：设 x 为与母体 X 同分布的随机变量

第 0 步：令 $j=0$(迭代次数)，设 $C^{(0)}=\{v_1^{(0)},v_2^{(0)},\cdots,v_k^{(0)}\}$ 为初始码本。

第 1 步：(划分)构造集合 $V^{(j)}=\{V_1^{(j)},V_2^{(j)},\cdots,V_k^{(j)}\}$，具体元素如下：

对 $1\leqslant i\leqslant k$，$V_i^{(j)}=\{x\mid\Delta(x,v_i)^{(j)}\leqslant\Delta(x,v_m^{(j)}),m=1,2,\cdots,k\}$

第 2 步：(期望)$C^{(j+1)}=\{v_1^{(j+1)},v_2^{(j+1)},\cdots,v_k^{(j+1)}\}$

对 $1\leqslant i\leqslant k$

$$v_i^{(j+1)}=\arg\min_v E[\Delta(x,v)\mid x\in V_i^{(j)}]=E[x\mid x\in V_i^{(j)}]$$

第 3 步：当 $(1-\Delta q(j+1)/\Delta q(j))$ 小于某个预先设定的阈值时，算法终止，否则置 $j=j+1$，返回第 1 步。

对离散数据集 $\chi_n=\{x_1,x_2,\cdots,x_n\}\subset R^d$，只需将上述算法第 2 步修改为：

第 2 步：$C^{(j+1)}=\{v_1^{(j+1)},v_2^{(j+1)},\cdots,v_k^{(j+1)}\}$，具体元素如下：

对 $1\leqslant i\leqslant k$

$$v_i^{(j+1)}=\arg\min_v\sum_{x\in V_i^{(j)}\cap\chi_n}\Delta(x,v)=1/|V_i^{(j)}|\sum_{x\in V_i^{(j)}\cap\chi_n}x$$

同时将第 3 步中的 $(1-\Delta q(j+1)/\Delta q(j))$ 换成 $(1-\Delta_n q(j+1)/\Delta_n q(j))$，余下相同。

1.2 自组织拓扑映射

竞争性过程：令 d 表示输入空间的维数，从输入空间中随机选择输入模式记为 $x=[x_1,x_2,\cdots,x_d]^T$，网络中每个神经元的突触值向量和输入空间的维数相同

$$w_j=[w_{j1},w_{j2},\cdots,w_{jd}]^T\quad j=1,2,\cdots,l$$

其中 l 为网络中神经元的总数。为了找到输入向量 x 与突触权值向量 w_j 的最初匹配，对于 $j=1,2,\cdots,l$，通过比较内积 w_j^Tx 并选择最大者。这里假定所有的神经元有相同的阈值，当阈值偏置时取负。这样，通过选择最大内积的 w_j^Tx 神经元，实际上决定了兴奋神经元的拓扑邻域中心的位置。

设 $w_{j0}=\arg\max_{1\leqslant j\leqslant 0}w_j^Tx$，那么 $w_{j0}^Tx=\max_{1\leqslant j\leqslant 0}w_j^Tx$，则

$j_0(x)=\arg\min_{1\leqslant j\leqslant l}|x-w_j|$，此时 $j_0(x)$ 标识最优匹配输入向量 x 的神经元，神经元 j_0 被称为输入向量 x 的神经元或获胜神经元。从而，激活模式的连续输入空间通过网络神经元之间的竞争过程映射到神经元的离散输出空间。

合作过程：从以上讨论了解到，获胜神经元位于合作神经元的拓扑邻域的中心，但问题的关键在于：如何定义一个在神经生物学意义上正确的拓扑邻域？

从神经生物学上讲，一组兴奋神经元的侧向相互作用是以有证据为基础的，一个点火的神经元倾向于激活它紧接的邻域内的神经元而不是与它隔得很远的神经元。这个观察引导我们对获胜神经元的拓扑邻域按侧向距离光滑地缩减。

设 h_{ji} 表示以获胜神经元 i 为中心得拓扑邻域，d_{ij} 表示获胜神经元 i 与兴奋神经元 j 的侧向距离。我们总是假定拓扑邻域 h_{ji} 是侧向距离 d_{ij} 的单峰函数，且满足下列两个不同的要求：

1. 拓扑邻域 h_{ji} 关于 $d_{ij}=0$ 定义的最大点是对称的，即在获胜神经元 i 处达到最大值。

2. 拓扑邻域 h_{ji} 的幅值随侧向距离 d_{ij} 的增加而单调下降，并且当 $d_{ij}\to\infty$ 时，其幅值趋向于零(对于收敛来说，这是必要条件)。

在通常单峰函数(即窗函数)中，Gauss 函数就是满足这些要求的一个 h_{ji} 的典型选择。

$$h_{ji}(x)=\exp(-d_{ji}^2/2o')$$

它是平移不变的(即不依赖于获胜神经元的位置)。o' 是拓扑邻域的"有效宽度"，它度量获胜神经元与兴奋神经元在学习过程中参与的程度。从生物学上看，Gauss 拓扑邻域比矩形形式的拓扑邻域更合适。它的应用使得 SOM 算法的收敛速度比矩形拓扑邻域更快(Lo et al. 1991, 1993; Erwin et al. 1992a)。

相对于邻域函数，神经元之间的合作必然要求拓扑邻域函数 h_{ji} 依赖获胜神经元 i 与兴奋神经元 j 在输出空间的侧向距离 d_{ij}，而不是依赖于原始输入空间的某种距离度量。对于一维网络而言，$d_{ij}=|j-i|$；而对于高维网络形，则定义 $d_{ij}^2=\|r_j-r_i\|^2$，其中离散向量 r_j 刻画神经元 j 的位置。

本文将采用与 Hastie[7] 相类似的窗函数(这不是本质性的)，并为了方便采取全局相同的邻域半径 o'。

自适应过程：作为特征映射自组织形式的最后一个过程，即突触自适应过程。为了使网络成为自组织的，必然要求神经元 j 的突触权值随输入向量 x 改变。

在 Hebb 学习假设中，突触权值随着前后突触的激活同时发生而增加。此方法非常适合联想学习。然而对于这里考虑的无监督学习，以 Hebb 假设的基本形式是不能令人满意的。理由是连接的改变仅发生在一个方向上，这样最终使所有的突触权值都趋于饱和。为此，须引入一个遗忘项 $g(y_j)w_j$ 来改变 Hebb 假设，其中 w_j 是神经元 j 的突触权值向量，

$g(y_j)$是响应 y_j 的正标量函数并满足 $g(0)=0$。突触权值向量 w_j 的增量更改为

$$\Delta w_j=\eta y_j x-g(y_j)w_j$$

其中 η 是算法的学习率。因此增量 Δw_j 是由 Hebb 项与遗忘项构成。通常最简单的选择是令 $g(y)=\eta y$，并且置 $y_j=h_{ji}(x)$，那么 $\Delta w_j=\eta h_{ji}(x)(x-w_j)$。

如果使用离散时间形式，那么便有如下差分形式 $\Delta w_j(n)=w_j(n+1)-w_j(n)$并且

$$w_j(n+1)=w_j(n)+\eta h_{ji(x)}(n)(x-w_j(n))$$

此式展示了网络中获胜神经元 i 的拓扑邻域中，获胜神经元 i 相对输入向输入向量 x 移动的作用。因此该算法导致在输入空间中特征映射的拓扑排序，即相邻神经元会有相似的突触权值向量。

1.3 主曲线

主曲线理论是主成分分析(PCA)理论的非线性推广，它使用曲线代替直线来描述数据点集，并光滑地通过数据集的"中央"。该曲线与回归曲线的差别在于数据集的坐标变量 x 和 y 是对称的。与线性主成分一样，获得曲线的关键点仍是数据点到曲线的最短距离。由此，给出如下定义：

定义 4 在随机分布密度函数 h 下，如果

$E(x|t_f(x)=t)=f(t)$，那么称曲线 f 是自相合的(或称主曲线)。

定义 5 对给定的数据分布或数据集，满足自相合(self-consistent)的光滑曲线，称为主曲线。

由定义 4 知，所谓自相合是指：在曲线上任取一点，收集所有投影在该点的数据，求它们的期望均值，其均值恰好与该点值一致。

2 基于自组织拓扑映射的主曲线学习算法

设数据集 $\chi_n=\{x_1,x_2,\cdots,x_n\}\subset R^d$，并假设 χ_n 的相关矩阵 $R=xx^T$ 的最大特征值不是重根，其中 $X=(x_1,x_2,\cdots,x_n)$ 为 $d\times n$ 矩阵。

设 $a=\frac{1}{n}\sum_{i=1}^{n}x_i=M(\chi_n)=0$，

$\Delta_n(f(t))=\frac{1}{n}\sum_{i=1}^{n}\Delta(x_i,f(t))$。

如果 $a=M(\chi_n)\neq 0$，可将数据集 χ_n 进行平移，置 $x'_i=x_i-a$，并令

$\chi_n'=\{x'_1,x'_2,\cdots,x'_n\}$，此时，有 $a'=M(\chi_n')=0$

因此总是可设 $M(\chi_n)=0$。以下我们总假设 $a=\frac{1}{n}\sum_{i=1}^{n}x_i=M(\chi_n)=0$。对于一般分布，基于 $E(X-E(X))=0$ 可知如上方法总可行。

算法 C(获取主曲线的算法)：

第 1 步：通过 RTB 算法求出第一主成分线 $f_0(t)$。

第 2 步：在主成份线上增加顶点，形成最初折线 $f'_k(t)$。

第 3 步：利用自组织拓扑映射进行自适应学习，调整顶点位置，形成新的折线 $f_k(t)$($f_k(t)$为通过第 ki 次增加顶点后形成的折线)。考虑误差，如果前后两次学习后的折线误差在允许范围内，则迭代停止，否则执行第 4 步。

第 4 步：利用向量量化器，增加折线顶点，形成折线 $f_{k+1}(t)$，转第 3 步重新调整顶点。

其中，第 1 步 RTB 算法可以参见作者以前发表的文章。

对算法的第 3 步，可以分解如下：

设 m 为自组织学习的迭代次数，k 为增加顶点的迭代次数。$f_m^{(k)}(t)$为经过 m 次迭代学习 k 次增加顶点后形成的折线。

第 3-1 步：计算每个观测值 $x_i\in\chi_n$ 在折线上的投影点 $y_i^{(k)}(m)$和相应的弧长参数 $t_{mi}^{(k)}$。

$$y_i^{(k)}(m)=\arg\min\{\|x_i-y\|\,|\,y\in f_m^{(k)}(t)\}$$

$$t_{mi}^{(k)}=\min\{t|\,\|f_m^{(k)}(t)-x_i\|=\min_{y\in f_m^{(k)}(t)}\|x_i-y\|\}$$

如果在折线上产生多个满足上述条件的投影点，则取最接近折线左端点的投影点。

第 3-2 步：选择窗函数。(窗函数的选择不唯一，主要是依据所处理问题的背景)

定义每个观测值 x_i 对于折线 $f_m^{(k)}(t)$的顶点 $v_{m,j}^{(k)}$的权值 $w_{(i,j)}^{(k)}$。$j=1,\cdots,N_k$，$i=1,\cdots,n$(其中，N_k 为折线顶点的个数，n 为观察值的个数)

$$w_{(i,j)}^{(k)}=\begin{cases}(1-(|t^{(k)}(m,j)-t_{mi}^{(k)}|^3)/\sigma)^{1/3} & \text{如}\ |t^{(k)}(m,j)-t_{mi}^{(k)}|<\sigma\\ 0 & \text{其他}\end{cases}$$

式中的 $t^{(k)}(m,j)$为第 m 次迭代时在第 k 条折线上第 j 个顶点的弧长。

第 3-3 步：自组织学习

定义新的调整后的折线顶点为：

$$v_{m+1,j}^{(k)}=v_{m,j}^{(k)}+\eta_k\left(\frac{\sum w_m^{(k)}(i,j)x_i}{\sum w_m^{(k)}(i,j)}-v_{m,j}^{(k)}\right)$$

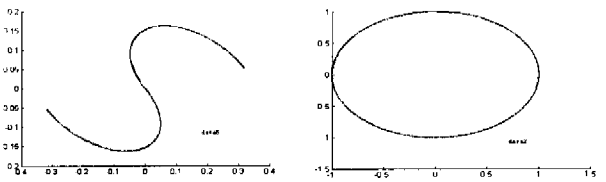
其中，m,k,j 的含义同上。η_k 为学习率参数，可定义为较小的常数(例如 $\eta_k=0.05$)。形成的新折线为 $f_{m+1}^{(k)}(t)=\{v_{m+1,j}^{(k)}|j=1,2,\cdots,N_k\}$。

第 3-4 步：控制自组织学习的迭代循环次数

如果 $\delta_{k,m+1}=1-\Delta_n(f_{m+1}^{(k)}(t))/\Delta_n(f_m^{(k)}(t))<\varepsilon$，则当 $k\geq\log_2 n$，则增加顶点的迭代停止，输出主曲线 $f(t)=f_k(t)$；否则，停止自组织学习迭代，并设 $f_k(t)=f_{m+1}^{(k)}(t)$，转第 4 步增加顶点；否则置 $m=m+1$，回到第 3-1 步，继续自组织学习的迭代。

3 算法分析与讨论

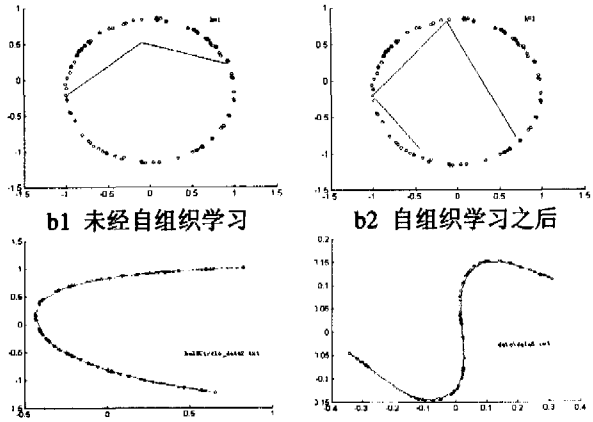
3.1 数据集相对密集或规范时自组织学习对算法结果的影响



组图(a)：密集数据集

上述一组图(a)是在未进行自组织学习的情况下，单纯利用向量量化器对数据集主曲线进行模拟时产生的结果。从组图中可以看出，无论数据集的形态是什么，只要数据点密集，向量量化器可以很好地取代自组织学习，反映数据集的变化形态。

下面一组图(b)表明：如果数据集中的数据稀疏但分布规范，则自组织学习只是在刚刚开始迭代时(例如 $k=1$ 时，k 为迭代次数)，由于主曲线顶点较少而显现出较大的优势(参见图 b1,b2)。但随着主曲线顶点以及迭代次数的不断增加，自组织学习对算法结果的影响逐渐减弱，进而对产生主曲线的作用不明显。



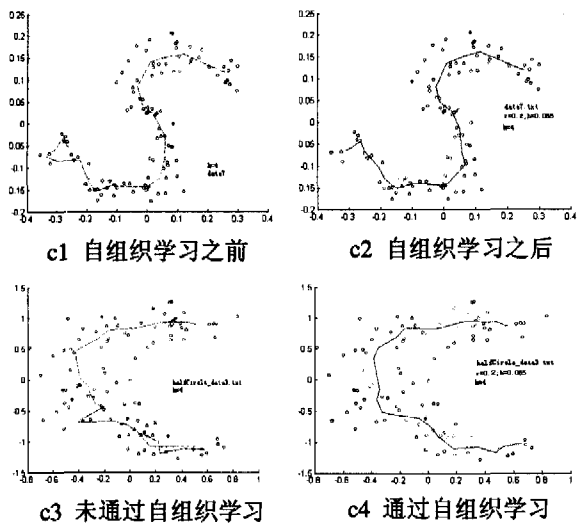
b1 未经自组织学习　b2 自组织学习之后

b3 未经自组织学习　b4 自组织学习之后

组图(b)：稀疏但变化规则的数据

综上所述，在数据集分布规范或数据集密集时，可以不经过自组织学习，即直接利用最优向量量化器对模拟主曲线顶点进行调整，并不断增加顶点，最终产生主曲线。这样做的好处在于避开了自组织学习所需的冗长的运行时间，从而可以大大提高算法的运行速度，算法的复杂度也将得到极大降低。

3.2 数据集相对稀疏时自组织学习对算法结果的影响



c1 自组织学习之前　c2 自组织学习之后

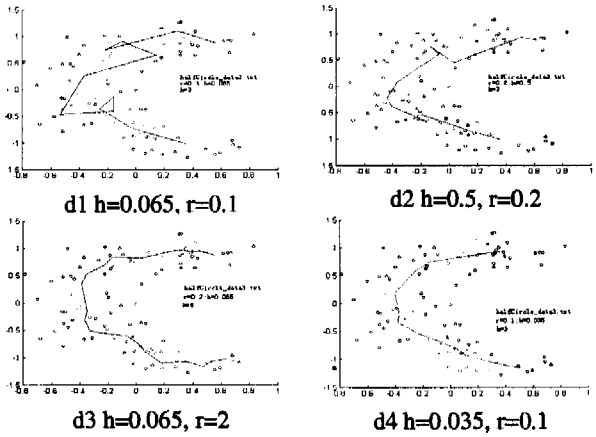
c3 未通过自组织学习　c4 通过自组织学习

组图(c)：稀疏且变化不规则的数据集

由组图(c)可以看出，当数据集稀疏且数据点分布不规则时，单纯依靠向量量化器来产生数据集的主曲线误差较大，难以正确反映数据集的特征变化趋势。通过选择适当的窗函数邻域参数和学习率参数，经过自组织学习后，一般情况下，主曲线在迭代过程中可以避免曲线随数据点较大的不规则扰动而产生较大的变化，并有一定的光滑性，从而使得主曲线更符合数据集的整体变化特征。

3.3 自组织学习算法中的相关参数设置对产生主曲线的影响

自组织学习算法中有两个关键参数：窗函数邻域参数(o')和学习率参数(η)(算法实现中分别用 r 和 h 表示)。邻域参数决定了窗函数的设置，亦即神经元对数据点的权值设置；而学习率参数决定了自组织学习算法的精度与效率。窗函数邻域参数与数据集的形态有很大关系，对较为稀疏的数据点集，不能取得太小，当然也不能取得过大，某种程度上取决于猜测和经验。学习率参数也是一样，一般与迭代的步数成反比。学习率越小，迭代精确度越高，但过小容易陷入亚稳定状态，因此也是经验参数。下面一组图(d)反映了这两个参数的变化带来的算法结果的变化。



d1 h=0.065, r=0.1　d2 h=0.5, r=0.2

d3 h=0.065, r=2　d4 h=0.035, r=0.1

组图(d)：自组织学习对参数的依赖性

结束语　通过以上比较我们知道自组织学习对最终结果产生明显的影响是有条件的，它依赖于离散数据集自身的特性和自组织学习算法中相关参数的选择。然而从智能化的角度看，尽管牺牲了一点速度，但总体效果有了改善，为提高识别率打下了基础。在我们本文所给出的算法基础上，如果增加基于三次样条的光滑化(样条光滑子[7])，即可得到 HS-型主曲线；如果将失真度 $\Delta(q)$ 变成 Kegl 等人[8]中带曲率惩罚项的失真度，便可得到 K-型主曲线。不过相对算法较现在复杂许多，对于较密集的数据集而言，精确度提高并不明显，有关结果以及与函数通用逼近器 RBF 网络和小波网络的比较将另文加以讨论。

参 考 文 献

1 Banfielld J D, Raftery A E. Ice floe identification in satellite images using mathematical morphology and clustering about principal curves [J]. Journal of the American Statistical Association, 1992, 87:7～16

2 Chang K, Ghosh J. Principal curve classufuer-a nonline approach to pattern classification [论文集]. In: IEEE International Joint Conf. on Neural Networks, 1998. 695～700

3 Chang K, Ghosh J. Principal curves for nonlinear feature extraction and classification[论文集]. In Applications of Artificial Neural Networks in Image Processing III, 1998, 3307:120～129

4 Delicado P. Principal curves and principal oriented points [Z]: [Technical Report 309]. Depart d'Economia I mpresa, Universitat Pomper Fabra, 1998

5 Delicado P. Another look at principal curves and surfaces [J]. Journal of Multivariate Analysis, 2001, 77(1):84～116

6 Haykin S. Neural Networks: A Comprehensive Foundation, 2nd Edition [M]. Pearson Education, Inc, 1999

7 Hestie T, Stuetzle W. Principal curves [J]. Journal of the American Statistical Association, 1989, 84(406): 502～516

8 Kegl B, Krzyzak A, et al. A polygonal line algorithm for constructing Principal curves. In: Proc. of Neural Information Processing Systems [J], Dever Colorado, USA, 1999. 501～507

9 Tibshirani R. Principal curves revisited [J]. Statistics and Computation, 1992, 2:183～190

10 张军平，王珏. 主曲线研究综述[J]. 计算机学报，2003, 26(2)

11 阮炯，顾凡及，蔡志杰. 神经动力学模型方法和应用[M]. 北京：科学出版社，2002