计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200500129

数据腐蚀对 GHTSOM 模型的优化

石 健[1] 莫 俊[2]
1 深圳市英维克信息技术有限公司 广东 深圳 518000
2 深圳市英维克软件技术有限公司 广东 深圳 518000
(gemingbanxie@163.com)

摘 要 聚类算法被广泛应用于模式识别、信息检索、图像处理,以及自然语言处理等领域,GCS 和 SOM 是两种常用的基于神经网络思想的聚类方式,很多学者在它们的基础上提出了不同的改进算法,GHTSOM(Growing Hierarchical Tree SOM)便是其中之一,对于数据分类较为清晰的应用场景效果良好,但不适用于干扰数据或者噪声数据较多的应用场景。利用图像处理中的腐蚀算法对 GHTSOM 算法进行优化,即在调用 GHTSOM 过程之前,先用腐蚀算法对数据进行处理,去除掉不同类别的数据交界位置处的干扰数据或者噪声数据,使不同类别数据之间出现较为明显的界限。为使表达更加直观,采用二维数据进行处理分析,结果表明,优化后的 GHTSOM 模型可有效避免由于类间局部连接造成的无法分类的问题,以及由于神经元过多所造成的误分类问题。

关键词:聚类;GHTSOM;数据腐蚀;数据处理;优化
中图法分类号 TP183

Optimization of GHTSOM Model by Data Corrosion

SHI Jian[1] and MO Jun[2]
1 Shenzhen Envicool Information and Technology Co.,Ltd.,Shenzhen,Guangdong 518000,China
2 Shenzhen Envicool Software Technology Co.,Ltd.,Shenzhen,Guangdong 518000,China

Abstract Clustering algorithm is widely used in pattern recognition,information retrieval,image processing and natural language processing. Two common clustering methods based on neural network are GCS and SOM. Many scholars have proposed different improved algorithms based on them. GHTSOM(Growing Hierarchical Tree SOM) is one of them. GHTSOM works well for applications where there is a clear classification of data,but it is not suitable for applications where there is a lot of noise or disturbing data. The corrosion algorithm in image processing is used to optimize the GHTSOM algorithm,that is,before calling the GHTSOM process,the data is processed by the corrosion algorithm to remove the interference data or noise data at the junction of different classes of data,making a distinction between different categories of data more obvious. To make the presentation more intuitive,two-dimensional datas are used. The results show that the optimized Ghsom model can effectively avoid the unclassifiable problems caused by local connections between classes and the misclassified problems caused by too many neurons.

Keywords Clustering,GHTSOM,Data corruption,Data processing,Optimize

1 引言

1.1 聚类算法

为了解决 GSC 算法聚类速度慢且难以表达的缺点,文献[1]提出了 TreeGCS 算法。文献[2]对 GCS 和 HiGS 模型中的节点插入、节点删除等策略进行了对比分析。HiGS 模型中的节点删除概率取决于上一次删除操作对网络产生的影响,加快了网络的自组织过程。文献[3]对 PRSOM 模型中 STVQ 算法的价值函数做了改进(加入一个正则项),使输出向量更加平滑。文献[4]采用了 P-SOM 算法来进行流形学习,成功地做到了学习数据的内在流形结构,并且具有一定的鲁棒性,能够在一定程度上有效容忍噪声的影响。LI 等提出了基于 one-hot 模型、核主元分析(KPCA)和自组织映射(SOM)网络的 RBC 系统智能故障诊断方法,并成功用于轨道交通列控 RBC 系统常见故障诊断,其准确率和处理效率得到进一步优化提升[5]。GHTSOM(Growing Hierarchical Tree SOM)是 Alberto Forti 在 SOM 模型的基础上改进的一种聚类模型算法[6]。

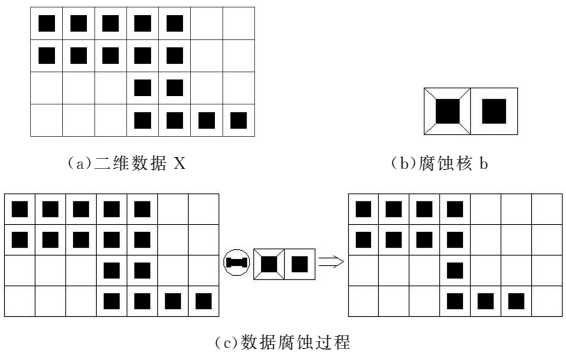
1.2 数据腐蚀算法

数学形态学最初被用于处理二值图像,是二值形态学的理论基础[7]。文献[7]将 SOM 模型与图像处理相结合,在城市道路卫星图片中识别主干道。膨胀运算、腐蚀运算、开运算、闭运算是数学形态学中的 4 种基本运算。其中,腐蚀运算用于削减边缘像素,使图形向内部收缩,还可消除噪声点[8]。其定义如下:假设 X,B 是域 ε^2 中的二值像素集合,X 被 B 腐蚀产生新的像素集合 Y。X,B,Y 中的像素点分别用 x,b,y 表示。则腐蚀被定义为[9]:

$$Y=X\Theta B=\{p\in\varepsilon^2:p=x+b\in X \text{ for every } b\in B\} \quad (1)$$

通信作者:莫俊(mushroom231@163.com)

例如：二维数据 X 和腐蚀核在图中分别表示为：



(a)二维数据 X　(b)腐蚀核 b　(c)数据腐蚀过程

图 1　数据腐蚀原理示意图

Fig. 1　Effect of corrosion operation on binary image

由上述例子可以看出，将二值图片进行腐蚀后，图片边缘被削减了一部分，而非边缘部分得以完全保留。

2　GHTSOM 聚类过程

GHTSOM 大体分为两个步骤：训练过程和聚类过程。

2.1　训练过程

该过程的目的是使数据样本映射到神经元向量空间，从而使神经元能够准确地体现数据样本的分布情况，主要步骤如下：

(1)在首层随机建立一个三角形全连接神经元(称之为 SOM 结构)，其结构如图 2 所示。

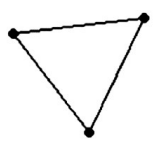


图 2　首层三角形全连接神经元结构示意图

Fig. 2　Structure of SOM with three neurons fully connected i the first level

(2)在首层中，对于每一个数据样本 A_i，分别计算其与各个神经元 N_j 之间的欧氏距离。距离最短的神经元为获胜神经元。

(3)首层 SOM 结构中的每个神经元在一定范围内随机生成 3 个子 SOM 结构，作为第 2 层，形成 SOM 结构树，如图 3 所示。图 3 中，实心圆表示首层神经元，空心圆表示第 2 层神经元。我们称上一层神经元(例如 A)为下一层神经元(例如 1，2，3)的父神经元，反之称为子神经元。

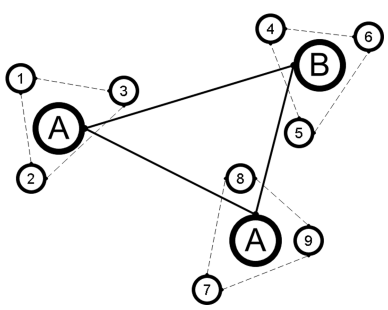


图 3　SOM 结构树生长示意图

Fig. 3　A tree of triangle SOMs with two levels

(4)利用每个数据样本对其所属的获胜神经元的子神经元及其邻域神经元进行训练，计算式如下：

$$N_j(n+1)=N_j(n)+\alpha(n)\cdot\lambda_{i,j}(n)\cdot(A_m-N_j(n)) \quad (2)$$

其中，A_m 表示第 m 个数据样本；N_j 表示第 j 个神经元，且其父神经元为 A_m 的获胜神经元；n 表示迭代步数；α 表示学习速率，随迭代步数的增加而增大；λ 为邻域方程，其表达式为：

$$\lambda_{i,j}(n)=\exp\left(-\frac{d_{i,j}^2}{2\sigma(n)^2}\right) \quad (3)$$

其中，$d_{i,j}$ 表示第 i 个神经元与第 j 个神经元的欧氏距离；σ 表示高斯函数标准差，且随迭代次数的增加呈指数下降趋势。

获胜神经元的邻域范围越大，学习速率也会越高，但同时最终的神经元分布也越不均匀。针对这一点，文献[10]用实验的方法做了验证。

(5)利用所有数据样本重新寻找第 2 层中的获胜神经元，并再次利用式(2)对神经元进行训练。

(6)评估第 2 层中所有神经元，如果获胜次数太少，则将其定义为叶子神经元，不再继续生长出子 SOM[6]。

(7)第 2 层中所有的非叶子神经元继续生长，产生下一层的 SOM，然后参照步骤 2－6 不断迭代，形成多层的 SOM 树。

2.2　聚类过程

该过程的目的是对当前层中所有神经元进行聚类，从而体现数据样本的分类情况，大体过程为：

(1)对于任意一个神经元 N_{j1}，计算所有使其获胜的数据样本 A_i 与 N_{j1} 的子神经元的欧氏距离。

(2)对于另外一个神经元 N_{j2}，计算 A_i 与其所有子神经元的欧氏距离。

(3)如果步骤(2)中所得到的最小距离小于步骤 1 中所得到的最大距离，则将 N_{j1} 与 N_{j2} 聚为一类。

2.3　标准 GHTSOM 模型聚类结果

为表达清晰，用二维数据对文献[6]中的 GHTSOM 模型进行实例化，原始数据如图 4 所示。

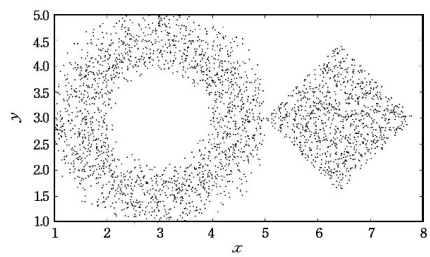


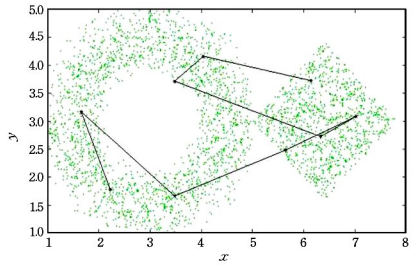
图 4　原始数据布局图

Fig. 4　Distribution map of input patterns

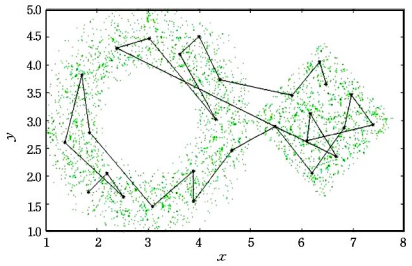
不难看出，图 4 中的数据分为两类：圆环和菱形。且两类数据之间存在着一定的连接。利用标准的 GHTSOM 模型对数据进行聚类，结果如图 5 所示，其中绿色小点表示数据样本，彩色大点表示神经元，用直线连接起来的所有神经元聚为同一类，不同类的神经元用颜色予以区分。

经过 9 层迭代后，依然没有将两类数据区分出来。由图可见，第 2－5 层中，所有数据始终被归为一类，这是由于两类数据之间存在局部连接所导致的，标准的 GHTSOM 模型算法无法忽视这种局部连接。从第 6 层开始，出现了误分类，这并不是我们想要的结果。之所以出现误分类，是因为神经元数量随着层数的增加而剧增，导致两个数据之间很小的间隙

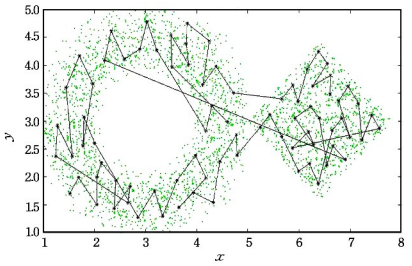
也被识别出来。



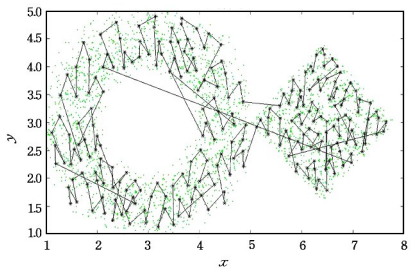
(a)第 2 层 GHTSOM 聚类结果



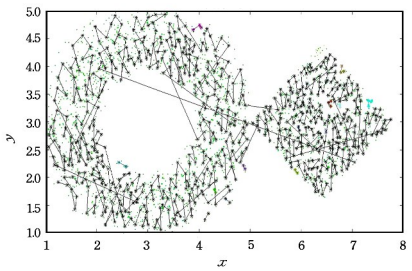
(b)第 3 层 GHTSOM 聚类结果



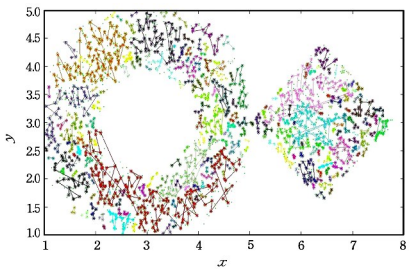
(c)第 4 层 GHTSOM 聚类结果



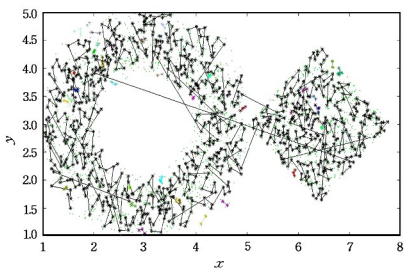
(d)第 5 层 GHTSOM 聚类结果



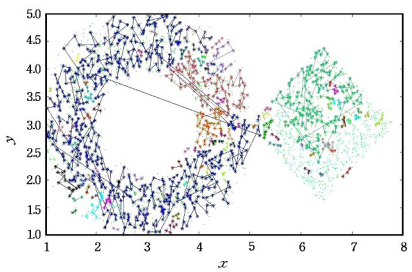
(e)第 6 层 GHTSOM 聚类结果



(f)第 7 层 GHTSOM 聚类结果



(g)第 8 层 GHTSOM 聚类结果



(h)第 9 层 GHTSOM 聚类结果

图 5 标准 GHTSOM 聚类结果(电子版为彩色)

Fig. 5 Clustering results of standard GHTSOM model

3 优化的 GHTSOM 聚类算法

通过对标准 GHTSOM 聚类算法结果的观察与分析,如果能够将两类数据之间的局部连接去除掉,聚类效果应该会有所提升,而图像处理中的腐蚀算法能够较好地满足这一点。

3.1 数据腐蚀过程

选取腐蚀核,核选取范围越大,被腐蚀掉的数据样本就越多。如果被腐蚀掉的数据样本过多,神经元就无法真实地反映数据样本的分布情况。相反,如果被腐蚀掉的数据量过少,则可能无法成功将两类数据连接处的干扰数据清除掉。通过对数据范围和密度的考虑,经过多次测试,最终选取边长为 0.17 的正方形作为腐蚀核的锚区域,向 4 个方向各延伸 0.34 的范围作为判断区域,如图 6 所示。

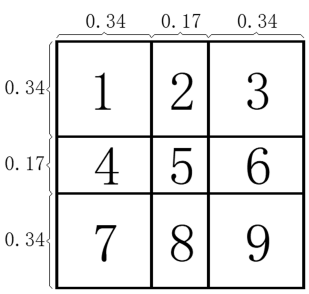


图 6 腐蚀核示意图

Fig. 6 Corrosion kernel

用腐蚀核顺序扫描数据样本空间,如果核中的 9 个区域均有数据存在,则锚区域 5 中的数据保留。腐蚀后的数据分布如图 7 所示。

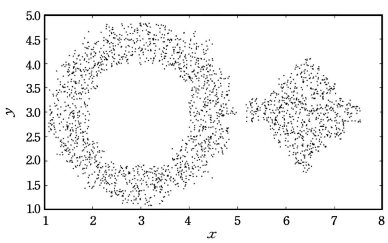


图 7 腐蚀后的数据分布

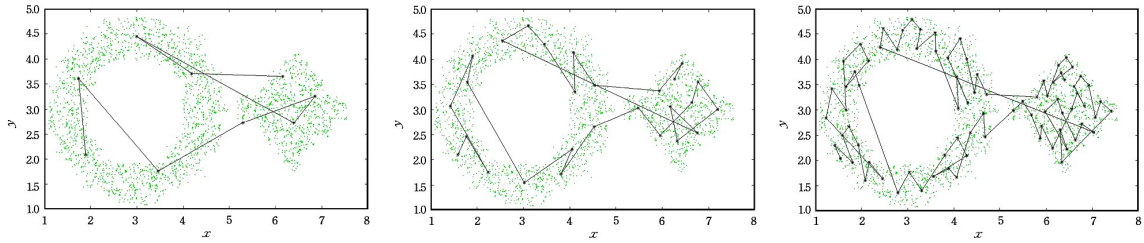
Fig. 7 Distribution map of input patterns after corrosion

经过腐蚀运算,两类数据的连接部分得到了明显的削减,同时数据边缘也略有削减。

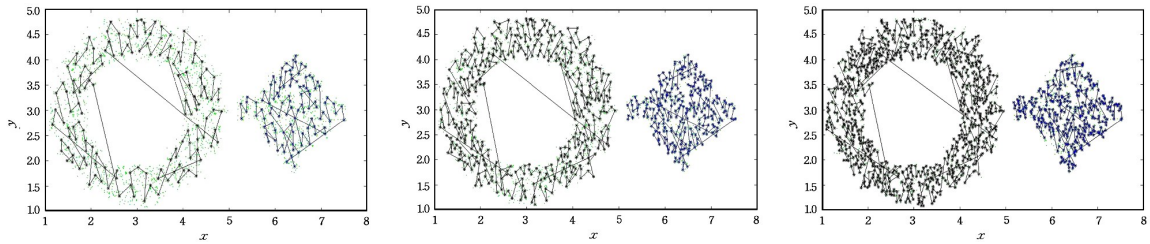
3.2 优化的 GHTSOM 聚类结果

将标准的 GHTSOM 算法应用于腐蚀后的样本数据,聚类结果如图 8 所示。由图 8 可知,从第 5 层开始就成功将两类数据区分开来,而且持续到第 9 层也没有出现误分类现象。这是因为聚类过程中充分利用腐蚀核的作用,将欧氏距离小于腐蚀核锚区域的数据样本也进行了聚类,有效避免了由于神经元过多导致的误分类现象。当两类数据样本成功区分出来之后,再对被腐蚀掉的那部分数据样本依据就近原则进行聚类运算。最终的全部数据样本分类结果如图 9 所示。

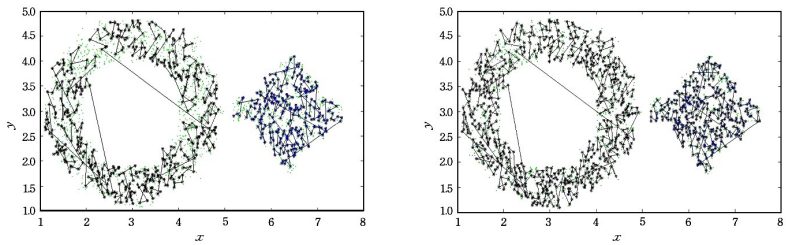
所有圆环形数据全部归为一类，用星号表示，所有菱形数据归为另一类，用圆圈表示。



(a)第 2 层优化后的 GHTSOM 聚类结果　(b)第 3 层优化后的 GHTSOM 聚类结果　(c)第 4 层优化后的 GHTSOM 聚类结果



(d)第 5 层优化后的 GHTSOM 聚类结果　(e)第 6 层优化后的 GHTSOM 聚类结果　(f)第 7 层优化后的 GHTSOM 聚类结果



(g)第 8 层优化后的 GHTSOM 聚类结果　(h)第 9 层优化后的 GHTSOM 聚类结果

图 8　优化后的 GHTSOM 聚类结果

Fig. 8　Clustering results of improved GHTSOM model

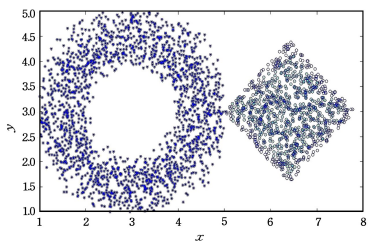


图 9　优化后的 GHTSOM 数据分类最终效果

Fig. 9　Clustering result of all the input patterns

结束语　利用腐蚀算法对数据进行处理，可有效消除类间连接，提高 GHTSOM 模型的聚类效果，同时有效避免了由于神经元过多所造成的误分类问题。但如何更有针对性地选择腐蚀核及优化腐蚀算法，仍然是一个值得深入研究的问题。

参 考 文 献

[1] HODGE V J, AUSTIN J. Hierarchical growing cell structures: TreeGCS [J]. IEEE Transactions on Knowledge and Data Engineering, 2001, 2(13): 207-218.

[2] BURZEVSKI V, MOHAN C K. Hierarchical growing cell structures[C]// Proceedings of International Conference on Neural Networks(ICNN'96). 1966: 1668-1663.

[3] WU S T, CHOW T W S. PRSOM: a new visualization method by hybridizing multidimensional scaling and self-organizing map [J]. IEEE Transactions on Neural Networks, 2005, 6(16): 1362-1380.

[4] 许洋. 基于 SOM 神经网络在流形学习中的研究与应用 [J]. 南方农机 2020, 51(9): 177-180.

[5] LI Y Q, LIN H X. Fault diagnosis method of train control RBC system based on KPCA-SOM network [J]. Journal of Measurement Science and Instrumentation, 2020, 11(2): 161-168.

[6] FORTI A, FORESTI G L. Growing Hierarchical Tree SOM: An Unsupervised neural network with dynamic topology [J]. Neural Networks, 2006(19): 1568-1580.

[7] YING H, WANG L Q, ZHAO X A. Automatic roads extraction from high-resolution remote sensing images based on SOM [C]// Circuits and Systems Society. 2010 Sixth International Conference on Natural Computation, 2010: 1194-1198.

[8] ZHANG Y B, ZHOU K. Study on automotive style recognition with the image erosion technology [C] // 2011 International Conference on Consumer Electronics. Communications and Networks(CECNet), 2011.

[9] ONKA M, HLAVAC V, BOYLE R. Image Processing, Analysis, and Machine Vision [M]// CL-Engineering. 2007.

[10] KOHONEN T. Self-Organized Formation of Topologically Correct Feature Maps [J]. Biological Cybernetics, 1982, 1(43): 59-69.

SHI Jian, born in 1988, master, intermediate engineer of thermal automation. His main research interests include artificial environment control algorithm design and data analysis.

MO Jun, born in 1985, bachelor, intermediate engineer of automation. His main research interests include design and test of control scheme for artificial environment.