第 37 卷 第 24 期
Vol.37 No.24

计 算 机 工 程
Computer Engineering

2011 年 12 月
December 2011

• 人工智能及识别技术 • 文章编号：1000—3428(2011)24—0198—02 文献标识码：A 中图分类号：TP393

基于几何距离摄动的局部线性嵌入算法

杨安平[1,2]，陈松乔[1]，胡 鹏[2]，莫禹均[2]

(1. 中南大学信息科学与工程学院，长沙 410083；2. 长沙理工大学电气与信息工程学院，长沙 410004)

摘 要： 传统局部线性嵌入(LLE)算法对近邻个数依赖性较强，不适用于处理稀疏数据源。针对该问题，提出一种基于几何距离摄动的 LLE 算法。通过线性块内的最大欧氏距离与测地距离之差构造几何摄动，描述流形数据的局部线性特性，对原始流形数据进行最大线性分块操作，保证局部模块的线性特性，并在每一个局部线性模块上应用 LLE 算法实现嵌入降维。实验结果表明，该算法能有效提高分类的平均准确率。

关键词： 特征提取；局部线性嵌入；流形学习；几何距离摄动；最大线性分块

Local Linear Embedding Algorithm Based on Geometric Distance Perturbation

YANG An-ping[1,2], CHEN Song-qiao[1], HU Peng[2], MO Yu-jun[2]

(1. School of Information Science and Engineering, Central South University, Changsha 410083, China;
2. School of Electrical & Information Engineering, Changsha University of Science and Technology, Changsha 410004, China)

【Abstract】 The traditional Local Linear Embedding(LLE) algorithm is sensitive for the number of nearest neighbors, and fails on sparse source data. In order to solve this problem, a local linear embedding algorithm based on the geometric distance perturbation is proposed. The local linear property of the manifold data is described by geometric distance perturbation. The original dataset is set into some maximal linear block according to the perturbation. The LLE is applied to this maximal linear patch to complete the embedding dimensional-reduction. Experimental results demonstrate this method can raise the average accurate rate.

【Key words】 feature extraction; Local Linear Embedding(LLE); manifold learning; geometric distance perturbation; Maximum Linear Block (MLB)

DOI: 10.3969/j.issn.1000-3428.2011.24.066

1 概述

特征提取是模式识别的关键技术之一，其任务是将原始数据空间映射到一个低维空间，在降低维数的同时，尽可能地保持原空间的重要信息。经过几十年的发展，特征提取技术取得长足的进步，传统的线性方法如主成分分析(Principal Component Analysis, PCA)、多维尺度分析(Multi-dimensional Scaling Analysis, MDS)等都取得了不错的效果[1]。

伴随现代信息技术的不断发展，原有的线性降维技术(如 PCA 等)已经不再符合现有数据分析的需求。自从 21 世纪初数据的流形结构被成功发掘后，流形学习的方法成为了当下主流的学习方法，相关的算法也不断涌现。其中旨在保持局部几何性质的局部线性嵌入(Locally Linear Embedding, LLE)算法因具有解析的整体解，不需要迭代运算；计算复杂度相对较小，容易执行的特点而受到了研究者的青睐，并已经成功应用于手写数字识别、人脸识别等实际问题[2]。

然而，在实际应用中发现，LLE 算法对近邻个数 k 的选取具有极强的依赖性，现有的大多数应用中，LLE 的参数 k 都通过经验选取，具有很大的局限性[3]。针对这个问题，本文提出一种基于几何距离摄动的局部线性嵌入算法。

2 局部线性嵌入

文献[2]提出了 LLE 算法。给定数据集 $\boldsymbol{X}=\{x_i\}_{i=1}^N \in R^D$，LLE 在保持数据点的局部邻域不变的同时，获得数据内在低维描述 $\boldsymbol{Y}=\{y_i\}_{i=1}^N \in R^d \ (d \ll D)$。

LLE 算法步骤如下：

Step1 对每个数据点 x_i 计算其近邻，可采用 k 近邻或 ε 邻域。

Step2 对每个数据点 x_i，由其邻域关系，计算线性重构系数 W_{ij}，使 $\left\|x_i-\sum_j W_{ij}x_j\right\|$ 最小，且 $\sum_j W_{ij}=1$；权重可直接由式(1)计算得到：

$$W_{ij}=\frac{\sum_k\left(G_{jk}^i\right)^{-1}}{\sum_{lm}\left(G_{lm}^i\right)^{-1}} \tag{1}$$

其中，$G_{jk}^i=\left(x_i-x_{ij}\right)\cdot\left(x_i-x_{ik}\right)$，$x_{ij}$、$x_{ik}$ 是 x_i 的近邻点。

Step3 计算 $\boldsymbol{Y}=\left(y_1,y_2,\cdots,y_n\right)$，使代价函数 $\phi(\boldsymbol{Y})=\sum_i\left\|\boldsymbol{Y}_i-\sum_j\boldsymbol{W}_{ij}\boldsymbol{Y}_j\right\|^2=trace\left(\boldsymbol{Y}^{\mathrm{T}}(\boldsymbol{I}-\boldsymbol{W})^{\mathrm{T}}(\boldsymbol{I}-\boldsymbol{W})\boldsymbol{Y}\right)$ 最小。

LLE 算法假定数据及所在的低维流形和从低维流形到高维观测空间的光滑嵌入映射在局部都是线性的，低维流形在观测空间中的像在局部也是线性的，观测空间中的每个数据点都可以用它的近邻点线性表示，且具有与在低维流形上的原像点相同的线性结构，即低维流形上的每个点用其近邻点

基金项目： 国家自然科学基金资助项目(61074018)

作者简介： 杨安平(1965－)，男，副教授、博士研究生，主研方向：图像处理，智能优化算法；陈松乔，教授、博士生导师；胡 鹏，硕士研究生；莫禹均，本科生

收稿日期： 2011-06-08 **E-mail：** yang_ap@yahoo.cn

线性表示的权重与它们在高维空间中线性表示的权重相同。

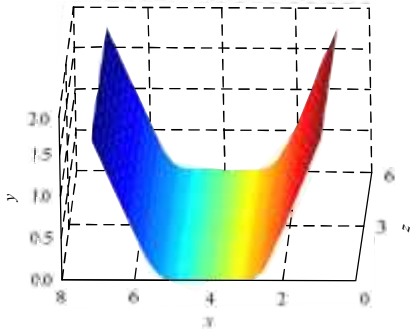
3　基于几何距离摄动的 LLE 算法

对于数据本身来说，低维流形和光滑嵌入映射的局部线性性质只是一种假设，在大部分情况下不能严格成立，另外，加上数据中的噪声等，这种线性一般都会有误差。尤其在样本分布稀疏的部分区域，k 个近邻所组成的局部邻域比在样本分布较密集的部分区域内由 k 个近邻所组成的局部邻域大得多，这时数据的局部线性特性得不到保证，极大程度上影响嵌入效果[4]。为此，本文通过线性块内的最大欧氏距离与测地距离之差构造的几何摄动，描述流形数据局部线性特性。

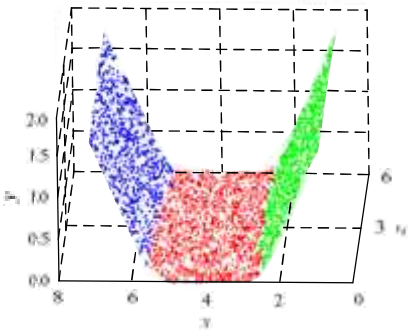
3.1　基于几何距离摄动的最大线性分块

从 LLE 算法步骤中可以看出，LLE 算法的关键在于原始数据空间的局部线性特性的保持。因为当 LLE 算法中某一点由其不在同一线性平面上的近邻点重构时，该算法就会失效，所以如果能够事先将原始数据划分成若干个局部线性块保证局部的线性特性，则能有效解决 LLE 算法失效的问题。为从非线性流形上构造出局部线性模型，研究者提出许多方法[5]。然而，这些方法大多采取一种基于迭代的聚类方法，这些方法有 2 个缺点：(1)目标类别的数目需要事先指定；(2)所提取的局部模块的线性度不能得到明确的保证。为了克服这 2 个问题，提出一种基于摄动的分块算法构造局部线性模型[6]。

最大线性块定义为流形曲面上一定线性限定条件下的最大局部线性块(Maximal Linear Block, MLB)。在几何直观上看，一定限定条件下线性块内的最大欧氏距离与测地距离之差构造的几何摄动能表示这样局部线性。在局部线性块的定义下，每一个 MLB 从一个点产生，然后逐渐扩大，直到保证线性性质的限制条件被打破，从而将原始数据在保证其线性条件的情况下划分成若干块。图 1 表示利用最大线性分块的思想，通过最大线性分块算法构造出的线性分块结构。



(a)3D “U” 型数据



(b)划分成 3 个 MLB 后的 “U” 型数据

图 1　最大线性分块算法在 3D “U” 型数据上的实例

3.2　最大线性分块算法

给定数据集 $\boldsymbol{X}=\{x_1,x_2,\cdots,x_N\}$，$\boldsymbol{x}_i\in R^D$ 为 D 维列向量，N 是样本个数。假定这些样本点来自于低维流形 M。将整个数据集 $\boldsymbol{X}$ 划分成不同的 MLB 块，用 C_i 表示：

$$\boldsymbol{X}=\bigcup_{i=1}^{m}C_i,\text{且}C_i\cap C_j=\varnothing\left(i\neq j,i,j=1,2,\cdots,m\right) \tag{2}$$

$$C_i=\left\{x_1^{(i)},x_2^{(i)},\cdots,x_{N_i}^{(i)}\right\}\ \left(\sum_{i=1}^{m}N_i=N\right) \tag{3}$$

其中，m 为 MLB 的个数；N_i 为块 C_i 中数据点的个数。

最大线性分块算法描述如下：

Step1 初始化设置，令 $i=1,C_i=\varnothing,X_T=\varnothing,X_R=\boldsymbol{X}$。$X_T$ 用于存放划分好的样本点集，X_R 用于存放暂未进行划分的样本点集。

Step2 当 $X_R\neq\varnothing$ 时：

(1)从 X_R 中随机选择一个种子点记为 $x_1^{(i)}$，令 $C_i=\left\{x_1^{(i)}\right\}$，则 $X_R=X_R-\left\{x_1^{(i)}\right\}$。

(2)对于 C_i 中的任一样本点 $x_n^{(i)}$：

考虑该点的每个 $k-NN$ 近邻点 x_c，如果同时满足下面 2 个条件：

1) $x_c\in X_R$。

2) $\left|D_G\left(x_c,x_n^{(i)}\right)-D_E\left(x_c,x_n^{(i)}\right)\right|<\theta\ \left(\text{for }\forall x_n^{(i)}\in C_i\right)$ (4)

则更新 $C_i=C_i\cup\{x_c\},X_R=X_R-\{x_c\}$。这里 D_G,D_E 分别为点对间的测地距离矩阵和欧氏距离矩阵。

(3)循环步骤 2)直到没有新的样本点可以更新加入至 C_i 中。

(4) $X_T=\bigcup_{j=1}^{i}C_j,X_R=X-X_T$，更新 $i\leftarrow i+1,C_i=\varnothing$。

值得注意的是，等式(4)中的阈值参数 θ 反映的是 MLB 的非线性度，也就是说，θ 值越大意味着局部模型个数越小，相应的线性偏差也就大，反之亦然。

3.3　最大线性分块下的局部嵌入算法

综合最大分块线性算法和 LLE 算法，得到本文算法主要步骤：

Step1 通过最大线性分块算法将原始数据集 $\boldsymbol{X}$ 划分为若干个子集 $C_1,C_2,\cdots,C_n$，其中，$X=\sum_{i=1}^{n}C_i$，且 $C_i\cap C_j=\varnothing,i\neq j$。

Step2 对于 Step 1 中得到的每一个子集 $C_i=\{x_1,x_2,\cdots,x_m\}$；对于 C_i 中的每一个点 x_i，运用式(1)计算权重 W_{ij}，其中，$G_{jk}^{i}=\left(x_i-x_{ij}\right)\cdot\left(x_i-x_{ik}\right)$，$x_{ij},x_{ik}$ 是 x_i 的近邻点而且 $x_{ij},x_{ik}\in C_i$。

Step3 计算：

$$\boldsymbol{Y}=\left(y_1,y_2,\cdots,y_n\right)=\min\phi\left(\boldsymbol{Y}\right)=\Sigma_i\left\|Y_i-\Sigma_j\boldsymbol{W}_{ij}\boldsymbol{Y}_j\right\|^2=$$
$$trace\left(\boldsymbol{Y}^{\mathrm{T}}\left(\boldsymbol{I}-\boldsymbol{W}\right)^{\mathrm{T}}\left(\boldsymbol{I}-\boldsymbol{W}\right)\boldsymbol{Y}\right)$$

4　实验结果与分析

为验证本文算法的有效性，将本文算法应用于 UCI 标准数据集，有 Glass、Iris、Wine 等数据集，如表 1 所示。

表 1　实验中所使用的数据集

数据集	数据点个数	维数	类别个数
Glass	345	6	2
Iris	150	4	3
Wine	178	13	3

(下转第 203 页)