基于核函数的稳健线性嵌入方法

徐雪松[1)] 宋东明[1)] 张 谔[1)] 许满武[2)] 刘凤玉[1)]

[1)](南京理工大学计算机科学与技术学院, 南京 210094) [2)](南京大学计算机科学与技术系, 南京 210093)

摘 要 LLE 算法是一种新的非监督学习方法,主要针对非线性降维问题。针对该算法存在的缺点,提出了一种基于核函数的稳健线性嵌入方法,该方法通过引进核函数来优化算法邻域点的求解;在特征空间中,修正权值矩阵 ***W***,进行降噪处理,经过推导,最终将实际的子空间计算归结为标准的特征值分解问题。采用最小近邻分类器估算识别率。在 Yale 人脸库以及 AT&T 人脸库的测试结果表明,在姿态、光照、表情、训练样本数目变化的情况下,改进的算法都具有较好的识别率。

关键词 流形学习 高维数据 维数约减 核函数

中图法分类号: TP391.41 **文献标识码**: A **文章编号**: 1006-8961(2009)06-1141-07

Robust Linear Embeding Based on a Kernel Function

XU Xue-song[1)], SONG Dong-ming[1)], ZHANG Xu[1)], XU Man-wu[2)], LIU Feng-yu[1)]

[1)](*Department of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing* 210094)

[2)](*Department of Computer Science and Technology, Nanjing University, Nanjing* 210093)

Abstract As a new unsupervised learning method, Local Linear Embedding algorithm (LLE) aims at reducing the nonlinear dimensionality. Since the local linear embedding method has many disadvantages, a new method, namely robust linear embedding method based on a kernel function, is presented to solve this problem. Firstly, the kernel function is utilized to adjust the Euclidean distance between data points, so the new method can improve the performance and the range of application of LLE. Secondly, the new method using the improved W is selected because it is insensitive to noise. It is shown that the actual computation of the subspace is reduced to a standard eigenvalue problem. The proposed method was tested and evaluated in the Yale face database and AT&T face database. Nearest neighborhood (NN) algorithm was used to construct classifiers. The experimental results showed that the improved algorithm has good performance when pose, lighting condition, face expression and train sample number change.

Keywords manifold learning, high dimensional data, dimensionality reduction, kernel function

1 引 言

人脸图像的维数通常是非常高的,实际上,人脸图像在这样高维空间中的分布很不紧凑,因而不利于分类。并且在计算上的复杂度也非常大。为得到较紧凑分布的人脸图像,Kirby 等人和 Turk 等人首次把主元分析的子空间思想引入到人脸识别中[1-2],并获得了较大的成功[3]。子空间分析方法随后引起了人们的广泛关注,从而成为了当前人脸识别的主要方法之一。子空间分析的思想是根据一定的性能目标来寻找线性或非线性的空间变换,把原始信号数据压缩到一个低维子空间,使数据在子空间中的分布更加紧凑,为更好地描述数据提供手段,同时计算的复杂度也得到了大大降低[4-5]。

由于线性子空间方法不足以有效地描述人脸图

收稿日期:2007-10-08;**改回日期**:2008-02-22

第一作者简介:徐雪松(1975~),男。南京理工大学计算机应用专业博士研究生。主要研究方向为模式识别、机器学习及信息安全。E-mail:xxs105@yahoo.cn

像中诸如光照、表情和姿态等复杂的非线性变化，而人脸空间更可能是一个非线性子空间，所以 Roweis 等人于 2000 年提出了 LLE(locally linear embedding)算法，是一种非线性降维方法[6]。LLE 算法可以学习到非线性流形的结构，但是它存在几个问题：(1)该算法要求样本在流形上是稠密采样[7]，导致在高维稀疏空间中采用欧氏距离公式失效；(2)在求解模型过程中可能出现的矩阵病态，会导致求解过程对于数据点噪声十分敏感[8]，使结果受噪声影响很大；(3)该算法的非线性特性往往使获得非线性映射只能定义于训练集上[9]，至于如何将其应用于检验集合尚待进一步研究，即若一个点不属于获得变换的训练集，则获得的变换就不能将这一点相应地映射到低维空间中。因此，本文提出一种改进的 LLE 流形算法即基于核函数的稳健线性嵌入方法。该算法基本思想是在 LLE 流形算法中引入核函数距离测度，以克服算法中由于高维稀疏空间所造成欧氏距离公式的点与点之间距离的对比性差等缺点，并从分析噪声对数据集局部特性的影响入手，通过修正权值矩阵 $\boldsymbol{W}$，可以很好地减弱噪声对降维过程的影响；然后在损失函数中引入一个线性变换矩阵，利用非线性核技巧在维数很高的空间里求解子空间，最终达到最佳降维效果。

2 LLE 算法

LLE 算法是一种依赖于局部线性的算法，它认为在局部意义下，数据结构是线性的，或者说局部意义下的点在一个超平面上，主要利用局部的线性来逼近全局的非线性，保持局部的几何结构不变，通过相互重叠的局部邻域来提供整体的信息，从而保持整体的几何性质[10-11]。

LLE 算法是映射数据 $A=\{\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n\}$，$\boldsymbol{a}_i\in\mathbf{R}^D$ 到数据集 $\boldsymbol{B}=\{\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_n\}$，$b_i\in\mathbf{R}^d(D>d)$。该算法主要包括 3 步：

(1)对高维空间中的每个样本点 $\boldsymbol{a}_i(i=1,2,\cdots,n)$，计算它和其他 $n-1$ 个样本点之间的距离，根据距离的大小，选择前 K 个与 $\boldsymbol{a}_i(i=1,2,\cdots,n)$ 最近的点作为其邻近点，常采用欧氏距离来度量两个点之间的距离，即 $d_{ij}=\|\boldsymbol{a}_i-\boldsymbol{a}_j\|$；

(2)对每个 $\boldsymbol{a}_i(i=1,2,\cdots,n)$，找到它的 K 个近邻点之后，计算该点和它的每个近邻点之间的权值 $w_j^{(i)}$，即最小化：

$$G(w)=\min\sum_{i=1}^{n}\left\|\boldsymbol{a}_i-\sum_{j=1}^{K}w_j^{(i)}\boldsymbol{a}_j\right\|^2 \tag{1}$$

式中，$\sum_{j=1}^{k}w_j^{(i)}=1$，如果 $\boldsymbol{a}_j(j=1,2,\cdots,k)$ 不是 $\boldsymbol{a}_i$ $(i=1,2,\cdots,n)$ 的近邻，则 $w_j^{(i)}=0$；

(3)根据高维空间中的样本点 $\boldsymbol{a}_i(i=1,2,\cdots,n)$ 和它的近邻 $\boldsymbol{a}_j(j=1,2,\cdots,K)$ 之间的权值 $w_j^{(i)}$ 来计算低维嵌入空间中的值 $\boldsymbol{b}_i$ 和 $\boldsymbol{b}_j$。由于在低维空间中尽量保持高维空间中的局部线性结构，而权值 $w_j^{(i)}$ 代表着局部信息，所以固定权值 $w_j^{(i)}$，使下面的损失函数最小化：

$$L(\boldsymbol{B})=\min\sum_{i=1}^{n}\left\|\boldsymbol{b}_i-\sum_{j=1}^{K}w_j^{(i)}\boldsymbol{b}_j\right\|^2=\mathrm{tr}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{B}) \tag{2}$$

式中，$\boldsymbol{M}=(\boldsymbol{I}-\boldsymbol{W})^{\mathrm{T}}(\boldsymbol{I}-\boldsymbol{W})$，$\boldsymbol{I}$ 为单位阵。

要求 $\sum_{i=1}^{n}\boldsymbol{b}_i=0$ 且 $\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{b}_i\boldsymbol{b}_i^{\mathrm{T}}=1$，以使 $L(\boldsymbol{B})$ 对平移、旋转和伸缩变化都具有不变性，使 $L(\boldsymbol{B})$ 最小化的解为矩阵 $\boldsymbol{M}$ 的最小几个特征值所对应的特征向量构成的矩阵 $\boldsymbol{B}$[12]。取 $\boldsymbol{M}$ 最小的 $m+1$ 个特征值对应的特征向量，去掉其中最小的特征值对应的特征向量，剩余的 m 个特征向量组成的矩阵就是低维空间中所得特征向量。

3 基于核函数的稳健线性嵌入方法

3.1 算法概述

假设输入空间数据集 $\boldsymbol{X}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n)$，$\boldsymbol{x}_k\in\mathbf{R}^D$，首先，该输入空间数据集被某种非线性映射 f 映射到特征空间(或是一个 Hibert 空间)得到 $f(\boldsymbol{x}_1)$，$f(\boldsymbol{x}_2)$，$\cdots$，$f(\boldsymbol{x}_n)$。基于核函数的稳健线性嵌入方法的目的是对特征空间中的数据点$(f(\boldsymbol{x}_1),f(\boldsymbol{x}_2),\cdots,f(\boldsymbol{x}_n))$进行降维，把它们映射到 d 维空间中形成新的数据点 $Y=(\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_n)$，其中 $d\ll D$。

(1)在特征空间中进行线性优化操作，对特征空间中的每个样本点 $f(\boldsymbol{x}_i)(i=1,2,\cdots,n)$，计算它和其他 $n-1$ 个样本点之间的距离，根据核函数距离的大小，选择前 $K^*(K^*=1,2,\cdots,n)$ 个与 $f(\boldsymbol{x}_i)(i=1,2,\cdots,n)$ 最近的点作为其邻近点，距离公式如下节详解，由参数 σ 决定 K^* 个近邻点紧致性调整；

(2)对每个 $f(\boldsymbol{x}_i)(i=1,2,\cdots,n)$，找到它的 K^* 个近邻点之后，将修正后的权值能量加入极小化目标函数使得

$$Q(\hat{\boldsymbol{W}}) = \min\left\{ \sum_{i=1}^{n} \left| f(\boldsymbol{x}_i) - \sum_{j=1}^{K^*} w_j^{(i)} f(\boldsymbol{x}_j) \right|^2 + \psi \|\boldsymbol{W}\|^2 \right\} \tag{3}$$

这样最终得到拓扑结构矩阵 $\tilde{\boldsymbol{W}}$,式中, $\sum_{j=1}^{k^*} w_j^{(i)} = 1$, 如果 $f(\boldsymbol{x}_j)(j=1,2,\cdots,k^*)$ 不是 $f(\boldsymbol{x}_i)(i=1,2,\cdots,n)$ 的近邻,则 $w_j^{(i)}=0$,Ψ 为系数;

(3)将线性变换矩阵引入 LLE 算法的损失函数中,并将其映射至特征空间,在特征空间中研究损失函数最小化问题;

(4)分解核矩阵 $\boldsymbol{k}$ 并计算矩阵 $\boldsymbol{M}$ 和 $\boldsymbol{Q}$,且 $\boldsymbol{K}=\boldsymbol{P\Lambda P}^{\mathrm{T}}$,$\boldsymbol{M}=(\boldsymbol{I}-\boldsymbol{W})(\boldsymbol{I}-\boldsymbol{W})^{\mathrm{T}}$,$\boldsymbol{Q}=\boldsymbol{P}^{\mathrm{T}}\boldsymbol{MP}$,其中 $\boldsymbol{I}$ 表示单位矩阵;

(5)求解下面的特征值分解问题得到向量 $\boldsymbol{R}_i$:

$$\boldsymbol{QR}_i = l_i^* \boldsymbol{R}_i \tag{4}$$

$l_i^*(i=1,2,\cdots,n)$ 为特征值,其中 $0<l_i<\cdots<l_d(d<n)$;

(6)计算向量 $\boldsymbol{\beta}_i$ 并对其归一化:

$$\boldsymbol{\beta}_i = \boldsymbol{P\Lambda}^{-1}\boldsymbol{R}_i$$
$$\boldsymbol{\beta}_i \leftarrow \boldsymbol{\beta}_i / \sqrt{\boldsymbol{\beta}_i^{\mathrm{T}} \boldsymbol{K} \boldsymbol{\beta}_i} \tag{5}$$

(7)用式(6)计算非线性降维

$$y_i^n = \sum_{j=1}^{n} \beta_j^n K_{i,j} \tag{6}$$

式中,$y_i{}^n$ 表示向量 $\boldsymbol{y}_i$ 的第 n 个分量。

3.2　算法的证明及分析

对步骤(1)中所用核距离公式推导如下:

在 Hibert 空间中,任意两点 $\boldsymbol{x}_1$,$\boldsymbol{x}_2$ 之间的点积为 $\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle$,定义特征空间中一点的范式为 $\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$。由于点积满足对称和双线形性质,因此,可以用范式定义特征空间中两点 $\boldsymbol{x}_1$,$\boldsymbol{x}_2$ 之间的距离为

$$\begin{aligned} \|\boldsymbol{x}_1 - \boldsymbol{x}_2\| &= \sqrt{\langle \boldsymbol{x}_1 - \boldsymbol{x}_2, \boldsymbol{x}_1 - \boldsymbol{x}_2 \rangle} \\ &= \sqrt{\langle \boldsymbol{x}_1, \boldsymbol{x}_1 \rangle - 2\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle + \langle \boldsymbol{x}_2, \boldsymbol{x}_2 \rangle} \end{aligned} \tag{7}$$

这样可以基于特征映射 f 来定义空间数据集中不同数据之间的关系,定义距离函数 f_d 为

$$f_d(\boldsymbol{d}_1, \boldsymbol{d}_2) = \|f(\boldsymbol{d}_1) - f(\boldsymbol{d}_2)\| = \sqrt{\langle f(\boldsymbol{d}_1), f(\boldsymbol{d}_1) \rangle - 2\langle f(\boldsymbol{d}_1), f(\boldsymbol{d}_2) \rangle + \langle f(\boldsymbol{d}_2), f(\boldsymbol{d}_2) \rangle} \tag{8}$$

但在很多情况下,由于特征空间维数很高,很难显示出一个数据映射到特征空间。然而,本文关心的仅仅是两个数据在特征空间内的像的点积,因此,不必显示计算出的两个像。

定义核函数来计算两个数据特征空间中的像的点积,它以两个数据点作为输入,输出是两个数据在特征空间中的像的点积:

$$F_f(\boldsymbol{d}_1, \boldsymbol{d}_2) = \langle f(\boldsymbol{d}_1), f(\boldsymbol{d}_2) \rangle \tag{9}$$

通过核函数可以重新定义式(7)为

$$f_d(\boldsymbol{d}_1, \boldsymbol{d}_2) = \sqrt{\langle F_f(\boldsymbol{d}_1, \boldsymbol{d}_1) - 2F_f(\boldsymbol{d}_1, \boldsymbol{d}_2) + F_f(\boldsymbol{d}_2, \boldsymbol{d}_2) \rangle} \tag{10}$$

可以通过核函数有效地计算特征空间中数据点之间的距离 f_d,例如高斯核函数:

$$F_{\mathrm{rb}}(\boldsymbol{d}_1, \boldsymbol{d}_2) = \exp\left(- \frac{\|\boldsymbol{d}_1 - \boldsymbol{d}_2\|^2}{\sigma^2} \right) \tag{11}$$

因此可得:

$$F_{\mathrm{rb}}(\boldsymbol{d}_1, \boldsymbol{d}_2) = \exp\left(- \frac{F_f(\boldsymbol{d}_1, \boldsymbol{d}_1) - 2F_f(\boldsymbol{d}_1, \boldsymbol{d}_2) + F(\boldsymbol{d}_2, \boldsymbol{d}_2)}{\sigma^2} \right) \tag{12}$$

核函数中的参数 σ 决定了对优化邻域紧致性的影响,参数 σ 的值越小优化邻域区域越紧致,但并非越小越好,太小的 σ 将导致无效的计算结果,该参数可根据实际情况进行选取。

对步骤(2)的推导如下:

由于 $\boldsymbol{W}$ 是由数据集 $f(\boldsymbol{X})$ 解出的,故当数据集 $\boldsymbol{X}$ 受噪声污染时,计算对噪声非常敏感,特别是当 $(f(\boldsymbol{X}))^{\mathrm{T}} f(\boldsymbol{X})$ 的特征值较小时,甚至得不到需要的结果。因此,权值矩阵 $\boldsymbol{W}$ 是否准确将直接影响低维嵌入的最终效果。下面研究当数据集受噪声污染时,权值 w_i 的变化情况。

令 $f(\boldsymbol{x}_0)$ 代表 $f(\boldsymbol{x}_i)$ $i=1,\cdots,n$,$f(\boldsymbol{x}_i)$ 代表相应的真实值,$\boldsymbol{U}(f(\boldsymbol{x}_0))$ 代表 $f(\boldsymbol{x}_0)$ 的邻域,设 $f(\boldsymbol{x}_1)$,$f(\boldsymbol{x}_2)$,$\cdots$,$f(\boldsymbol{x}_k) \in \boldsymbol{U}(f(\boldsymbol{x}_0))$,则

$$f(\boldsymbol{x}_0) = \sum_{i=1}^{k^*} \boldsymbol{W}^i f(\boldsymbol{x}_i) \qquad \sum_{i=1}^{k^*} \boldsymbol{W}^i = 1$$

令 $\hat{f}(\boldsymbol{x}_i) = f(\boldsymbol{x}_i) + d_i (i=0,1,2,\cdots,k)$ 代表相应的噪声影响点,以及相应的

$$\hat{f}(\boldsymbol{x}_0) = \sum_{i=1}^{k^*} \hat{\boldsymbol{W}}^i \hat{f}(\boldsymbol{x}_i) \qquad \sum_{i=1}^{k^*} \hat{\boldsymbol{W}}^i = 1$$

若再令

$$f(\boldsymbol{X}^0) = (f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \cdots, f(\boldsymbol{x}_k))$$

$\hat{f}(\boldsymbol{X}^0) = (\hat{f}(\boldsymbol{x}_1), \hat{f}(\boldsymbol{x}_2), \cdots, \hat{f}(\boldsymbol{x}_k))$,以及

$\boldsymbol{W} = (\boldsymbol{W}^1, \boldsymbol{W}^2, \cdots, \boldsymbol{W}^k)^{\mathrm{T}}$,$\hat{\boldsymbol{W}} = (\hat{\boldsymbol{W}}^1, \hat{\boldsymbol{W}}^2, \cdots,$

$\hat{\boldsymbol{W}}^k)^{\mathrm{T}}$ 于是有

$$f(\boldsymbol{x}_0)=f(\boldsymbol{X}^0)\boldsymbol{W},\ \hat{f}(\boldsymbol{x}_0)=\hat{f}(\boldsymbol{X}^0)\hat{\boldsymbol{W}}$$

式中，$\boldsymbol{X}^0$ 代表 $\boldsymbol{x}_0$ 点的邻域矩阵。

定理 在上述记号下，各点噪声之间，不同真实值之间，以及 $\hat{\boldsymbol{W}}$ 与噪声之间是相互独立的，各点噪声是同均值(0)，同方差的，并且记 $\delta\boldsymbol{W}=\hat{\boldsymbol{W}}-\boldsymbol{W}$，有如下判别式：

$$E\|\hat{\boldsymbol{W}}\|^2\geqslant E\|\delta\boldsymbol{W}\|^2\frac{\lambda_{\min}l}{k^*(k^*+1)\rho^2} \tag{13}$$

式中，$\|\cdot\|$取为欧几里德范数，$l=\mathrm{rank}(f(\boldsymbol{X}^0))$，$\lambda_{\min}$ 为 $(f(\boldsymbol{X}^0))^{\mathrm{T}}f(\boldsymbol{X}^0)$ 的最小非零特征值，d^i 代表 $\boldsymbol{d}$(距离向量)的第 i 个分量，

$$\rho^2=\sum_{i=1}^{k^*}\rho_i^2,\rho_i^2=\mathrm{Var}(d^i)\ (i=1,2,\cdots,k) \tag{14}$$

证明：由 $\hat{f}(\boldsymbol{x}_i)=f(\boldsymbol{x}_i)+d_i(i=0,1,\cdots,k)$ 可见

$$\hat{f}(\boldsymbol{x}_0)=\hat{f}(\boldsymbol{X}^0)\hat{\boldsymbol{W}}+\sum_{i=1}^{k^*}\hat{\boldsymbol{W}}^i d^i\Rightarrow f(\boldsymbol{x}_0)=f(\boldsymbol{X}^0)\hat{W}+\sum_{i=1}^{k^*}\hat{\boldsymbol{W}}^i(d^i-d^0)$$

进一步有

$$f(\boldsymbol{X}^0)(\hat{\boldsymbol{W}}-\boldsymbol{W})=\sum_{i=1}^{k^*}\hat{\boldsymbol{W}}^i(d^0-d^i),$$

$$\sum_{i=1}^{k^*}\hat{\boldsymbol{W}}^i=\sum_{i=1}^{k^*}\boldsymbol{W}^i=1$$

即 $$f(\boldsymbol{X}^0)\delta\boldsymbol{W}=\sum_{i=1}^{k^*}\hat{\boldsymbol{W}}^i(d^0-d^i) \tag{15}$$

由于噪声是独立的，则有

$$\begin{aligned}&E(\delta\boldsymbol{W}^{\mathrm{T}}(f(\boldsymbol{X}^0)^{\mathrm{T}}f(\boldsymbol{X}^0)\delta\boldsymbol{W})\\&=E\Big[\sum_{i=1}^{k^*}\hat{\boldsymbol{W}}^i(d^0-d^i)^{\mathrm{T}}\sum_{i=1}^{k}\hat{\boldsymbol{W}}^i(d^0-d^i)\Big]\\&=\sum_{i=1}^{k^*}E(\boldsymbol{W}^i)^2\rho^2+\sum_{i=1}^{k^*}E(\hat{\boldsymbol{W}}^i\hat{\boldsymbol{W}}^j)\rho^2\leqslant\\&(k^*+1)\rho^2\sum_{i=1}^{k^*}E(\boldsymbol{W}^i)^2\\&=(k^*+1)\rho^2E\|\hat{\boldsymbol{W}}\|^2\qquad(j=1,\cdots,k)\end{aligned} \tag{16}$$

记 $(f(\boldsymbol{X}^0))^{\mathrm{T}}f(\boldsymbol{X}^0)$ 的正交分解为

$$(f(\boldsymbol{X}^0))^{\mathrm{T}}f(\boldsymbol{X}^0)=\boldsymbol{S}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{S}$$

式中，$\boldsymbol{S}$ 为正交阵，$\boldsymbol{\Lambda}=\begin{bmatrix}\lambda_1&\cdots&0\\\vdots&&\vdots\\0&\cdots&\lambda_k\end{bmatrix}$。

由于 $\mathrm{rank}(f(\boldsymbol{X}^0))^{\mathrm{T}}f(\boldsymbol{X}^0))=\mathrm{rank}(f(\boldsymbol{X}^0))$，所以

$$E(\delta\boldsymbol{W}^{\mathrm{T}}(f(\boldsymbol{X}^0))^{\mathrm{T}}f(\boldsymbol{X}^0)\delta\boldsymbol{W})=E(\delta\boldsymbol{W}^{\mathrm{T}}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{S}\delta\boldsymbol{W})\geqslant\lambda_{\min}\sum_{i=1,\lambda_i=0}^{k}E(\delta\boldsymbol{W}^i)^2 \tag{17}$$

通过对 $\boldsymbol{S}$ 进行行初等变换(改变两行的位置)，然后计算相应的式(17)，将所得结果相加可得

$$E(\delta\boldsymbol{W}^{\mathrm{T}}f(\boldsymbol{X}^0)^{\mathrm{T}}f(\boldsymbol{X}^0)\delta\boldsymbol{W})=E(\delta\boldsymbol{W}^{\mathrm{T}}\boldsymbol{S}^{\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{S}\delta\boldsymbol{W})\geqslant\lambda_{\min}E\|\delta\boldsymbol{W}\|^2\frac{1}{k^*} \tag{18}$$

其中，$\lambda_{\min}=\min_{1\leqslant i\leqslant k^*}\{\lambda_i>0\}$。

结合式(16)可知

$$E\|\delta\boldsymbol{W}\|^2\leqslant\frac{k^*(k^*+1)}{\lambda_{\min}l}\sum_{i=1}^{k^*}E(\delta\boldsymbol{W}^i)^2 \tag{19}$$

即

$$E\|\hat{W}\|^2\geqslant E\|\delta\boldsymbol{W}\|^2\frac{\lambda_{\min}l}{k^*(k^*+1)\rho^2}$$

由上述定理可知，在邻域大小 k^* 已知情况下，$\boldsymbol{W}$ 的误差主要由 3 个因素决定：(1)噪声的影响，即 $\boldsymbol{d}$ 的大小；(2)邻域的影响，即 $\lambda_{\min}$ 和秩 l 的大小；(3)权值能量的 $\|\hat{\boldsymbol{W}}\|$ 的大小。因素(1)的影响来自于数据采样本身，因素(2)的影响可通过步骤 1 改变邻域来调整，本文主要通过修正因素(3)即是通过减少权值能量达到减弱噪声影响的目的。

步骤(3)至步骤(7)推导如下：

LLE 算法中式(2)

$$L(\boldsymbol{B})=\min\sum_{i=1}^{n}\Big\|\boldsymbol{b}_i-\sum_{j=1}^{K}w_j^{(i)}\boldsymbol{b}_j\Big\|^2=\mathrm{tr}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{B})$$

最小化损失函数的约束项为 $\boldsymbol{B}\boldsymbol{B}^{\mathrm{T}}=\boldsymbol{I}$。其中 $\mathrm{tr}(\cdot)$ 代表对矩阵求迹的算子。然而该最小化损失函数仅仅依赖于权值 w，其输出 $\boldsymbol{b}$ 没有显示地与输入点 $\boldsymbol{a}$ 相关联，也就不能产生便于应用的、可延展的变换函数。为了克服这一问题，把下面的线性变换矩阵引入最小化损失函数中：

$$\boldsymbol{B}=\hat{\boldsymbol{S}}^{\mathrm{T}}\boldsymbol{X}\quad\text{或}\quad\boldsymbol{b}_i=\hat{\boldsymbol{S}}^{\mathrm{T}}\boldsymbol{X}_i \tag{20}$$

其中，$\boldsymbol{S}=\{\boldsymbol{s}_1,\cdots,\boldsymbol{s}_d\}$ 于是有

$$\begin{aligned}L(\boldsymbol{B})&=\mathrm{tr}(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{B})=\mathrm{tr}[(\hat{\boldsymbol{S}}^{\mathrm{T}}\boldsymbol{X})\boldsymbol{M}(\hat{\boldsymbol{S}}^{\mathrm{T}}\boldsymbol{X})^{\mathrm{T}}]\\&=\mathrm{tr}[\hat{\boldsymbol{S}}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\mathrm{T}})\hat{S}]\end{aligned} \tag{21}$$

st

$$\hat{\boldsymbol{S}}^{\mathrm{T}}\boldsymbol{M}(\hat{\boldsymbol{S}}^{\mathrm{T}}\boldsymbol{X})^{\mathrm{T}}=\hat{\boldsymbol{S}}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})\hat{\boldsymbol{S}}=\boldsymbol{I} \tag{22}$$

利用 Lagarange 乘子求解上述受约束的最小化问题：

$$L(\hat{\boldsymbol{S}})=\hat{\boldsymbol{S}}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{M}\boldsymbol{X}^{\mathrm{T}})\hat{\boldsymbol{S}}+l_i^*(\boldsymbol{I}-\hat{\boldsymbol{S}}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\hat{\boldsymbol{S}}) \tag{23}$$

使得对于 $\hat{\boldsymbol{S}}$ 的梯度为 0，得到

$$(\boldsymbol{XMX}^{\mathrm{T}})\hat{\boldsymbol{S}} = l_i^{*}(\boldsymbol{XX}^{\mathrm{T}})\hat{\boldsymbol{S}} \tag{24}$$

或者

$$(\boldsymbol{XMX}^{\mathrm{T}})\boldsymbol{s} = l_i^{*}(\boldsymbol{XX}^{\mathrm{T}})\boldsymbol{s} \tag{25}$$

现在开始在高维特征空间 F 中研究上述最小化问题,在式(25)中引入非线性映射函数 f:

$$[f(\boldsymbol{X})\boldsymbol{M}(f(\boldsymbol{X}))^{\mathrm{T}}]\boldsymbol{s} = l_i^{*}[f(\boldsymbol{X})(f(\boldsymbol{X}))^{\mathrm{T}}]\boldsymbol{s} \tag{26}$$

因为特征向量可以用 F 中元素的线性组合表示,所以存在系数 $\boldsymbol{\beta}_i(i=1,\cdots,n)$ 使得:

$$\boldsymbol{s} = \sum_{i=1}^{n}\boldsymbol{\beta}_i f(\boldsymbol{x}_i) \tag{27}$$

把式(27)代入式(26)中并且两边乘以 $[f(\boldsymbol{x}_j)]^{\mathrm{T}}$ 得到:

$$[f(\boldsymbol{x}_j)\cdot f(\boldsymbol{x}_1),\cdots,f(\boldsymbol{x}_j)\cdot f(\boldsymbol{x}_n)]\boldsymbol{M}\sum_{i=1}^{n}\boldsymbol{\beta}_i\cdot \begin{bmatrix} f(\boldsymbol{x}_1)\cdot f(\boldsymbol{x}_i) \\ \vdots \\ f(\boldsymbol{x}_n)\cdot f(\boldsymbol{x}_i) \end{bmatrix} = l^{*}[f(\boldsymbol{x}_j)\cdot f(\boldsymbol{x}_1),\cdots,f(\boldsymbol{x}_j)\cdot f(\boldsymbol{x}_n)]\cdot \sum_{i=1}^{n}\boldsymbol{\beta}_i\cdot \begin{bmatrix} f(\boldsymbol{x}_1)\cdot f(\boldsymbol{x}_i) \\ \vdots \\ f(\boldsymbol{x}_n)\cdot f(\boldsymbol{x}_i) \end{bmatrix} \tag{28}$$

定义核矩阵 $\boldsymbol{K}_{n\times n}$(它的元素为)$K_{i,j} = (f(\boldsymbol{x}_i))^{\mathrm{T}}f(\boldsymbol{x}_i) = f(\boldsymbol{x}_i)f(\boldsymbol{x}_i)$ 和向量 $\boldsymbol{\beta} = [\boldsymbol{\beta}_1,\cdots,\boldsymbol{\beta}_n]$ 上式可写成更加紧凑的形式:

$$\boldsymbol{KMK\beta} = l_i^{*}\breve{\boldsymbol{K}}\boldsymbol{\beta} \tag{29}$$

上式计算可归结为广义特征值分解问题,如果矩阵 $\breve{\boldsymbol{K}}$ 可逆,则式(29)的广义特征值分解可以转换成标准特征值分解问题。只需在公式两边同时乘以逆矩阵 $(\breve{\boldsymbol{K}})^{-1}$。但实际情况是 $\breve{\boldsymbol{K}}$ 常常为奇异矩阵,为了避免此问题对式(29)继续推导:

在式(29)两边同时乘 $\boldsymbol{\beta}^{\mathrm{T}}$,得到

$$\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{KMK\beta} = l_i^{*}\boldsymbol{\beta}^{\mathrm{T}}\breve{\boldsymbol{K}}\boldsymbol{\beta} \tag{30}$$

对 $\boldsymbol{K}$ 实施特征值分解:

$$\boldsymbol{k} = \boldsymbol{P\Lambda P}^{\mathrm{T}} \tag{31}$$

其中,$\boldsymbol{P}$ 是正交矩阵而 $\boldsymbol{\Lambda}$ 是对角矩阵,把式(31)代入式(30),有

$$(\boldsymbol{\Lambda P}^{\mathrm{T}}\boldsymbol{\beta})^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{MP}(\boldsymbol{\Lambda P}^{\mathrm{T}}\boldsymbol{\beta}) = l_i^{*}(\boldsymbol{\Lambda P}^{\mathrm{T}}\boldsymbol{\beta})^{\mathrm{T}}(\boldsymbol{\Lambda P}^{\mathrm{T}}\boldsymbol{\beta}) \tag{32}$$

其中,应用了 $\boldsymbol{PP}^{\mathrm{T}} = \boldsymbol{I}$ 这一关系。

进一步,令 $\boldsymbol{R} = \boldsymbol{\Lambda P}^{\mathrm{T}}\boldsymbol{\beta}$ 从而 $\boldsymbol{\beta} = \boldsymbol{P\Lambda}^{-1}\boldsymbol{R}$,式(32)可简化为 $\boldsymbol{R}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{MPR} = l_i^{*}\boldsymbol{R}^{\mathrm{T}}\boldsymbol{R}$ 因此,下式成立

$$\boldsymbol{P}^{\mathrm{T}}\boldsymbol{MPR} = l_i^{*}\boldsymbol{R} \tag{33}$$

最后定义 $\boldsymbol{Q} = \boldsymbol{P}^{\mathrm{T}}\boldsymbol{MPR}$,于是有

$$\boldsymbol{QR} = l_i^{*}\boldsymbol{R} \tag{34}$$

注意到式(34)就是步骤(5)中的式(4)。同时在步骤(6)中对 $\boldsymbol{\beta}_i$ 的归一化,用来保证与之对应的向量 $\boldsymbol{R}_i$ 在 F 空间里归一的。联立式(20)和(27)就可以得到步骤(7)中的式(6)。至此,完成了所有推导。

4　算法的实现及其实验结果的评估

将本文提出的基于核函数的稳健线性嵌入方法,在 Yale 人脸库以及 AT&T 人脸库上分别进行了测试。Yale 人脸库用来测试在表情和光照变化的情况下算法的性能。AT&T 人脸库用来测试训练样本数目变化时算法的识别情况。图 1 是两个人脸库的图片示例。

图 1　AT&T 及 Yale 人脸库的人脸照片

Fig. 1　Sample images for one subject of AT&T and Yale database

Yale 人脸库包括 15 个人,每人 11 张 195×231 的灰度照片。这些照片面部表情、面部细节、光照等都有着不同程度的变化。在预处理阶段,将 Yale 人脸库中所有图片进行裁剪,归一化成 100×100 的标准人脸图像。在 Yale 库中,将每人的前 5 幅人脸图像样本作为训练样本,后 6 幅图像作为测试样本,这样训练样本总数为 75 个,测试样本总数为 90 个。AT&T 人脸库包括 40 个人,每人 10 张 112×92 的灰度照片。照片中人脸的位置和尺度都有不同程度的变化。在 AT&T 人脸库中每人随机选取 5 张照片用于训练,其余的用于测试,一共有 200 个训练样本和 200 个测试样本。我们将本文提出的算法分别与 Eigenface(PCA)和 LLE 算法进行比较,采用最小近邻分类器估算识别率。

在两个人脸数据库上的对比实验中采用"留一法"验证策略,该验证策略实际上是交叉验证的极端情况。这种方法的基本思路是:假定有 n 个初始样本,每次拿出其中一个样本做测试,其余 $n-1$ 个做训练样本,这样的过程重复 n 次,结果每个样本都

有一次作为测试样本。其中本文用 $n-1$ 个样本构建一个分类器,再用另外一个样本做测试,得到一个准确度,0 或者 100%,因为这样的过程做了 n 次,因此最后的准确度值为 n 次的平均值。这个方法有两个明显的好处:第一,它尽可能的使用了所有能用的数据来进行训练;第二,这个方法具有确定性,不需要像交叉验证那样要用随机函数来选择训练集与测试集的划分。

对于 RLEBKF 算法采用的高斯核

$$F_{\mathrm{rb}}(\boldsymbol{d}_1,\boldsymbol{d}_2)=\exp(-\|\boldsymbol{d}_1-\boldsymbol{d}_2\|^2/\sigma^2)$$

核函数中的参数 σ 决定了对优化邻域紧致性的影响。表 1 表示在 AT&T 人脸库及 Yale 人脸库上,RLEBKF 算法选取不同参数 σ^2 的值的识别精度。

表 1 核函数中的 σ 与优化区域正确识别率之间的关系

Tab. 1 Relationship between σ of kernel function and right recognition rates in the optimize region

参数 σ^2	0.5	0.3	0.1	0.05	0.03
AT&T 人脸库正确识别率	65	72	79	85	87.5
Yale 人脸库正确识别率	68	74	80	84.33	87

从表中可以看出,随着 σ^2 的减小,优化区域识别精度逐渐增大,当 $\sigma^2=0.03$ 时,识别精度已经达到 87% 以上。

在实验中可以看出,原 LLE 算法在保持原数据集特性的同时,由于数据集受到了噪声的污染,使得结果产生了一定的厚度,即将本征维数得到了增加,这显然有悖于处理的初衷。相反,本文算法则在很好地保持数据集特性的基础上,抑制了噪声的影响,可更加清晰地揭示数据集的本征结构。

表 2 表示在 Yale 人脸库及 AT&T 人脸库中,比较了 3 种不同识别方法可以达到的最佳识别率 $R_{\max}$ 和所需的特征维数 D_{im}。可以看出,在 Yale 人脸库及 AT&T 人脸库中 PCA 的识别率最低;LLE 虽然识别率较高,但是特征维数分别达 42×10 和 54×14;而本文方法分别只需要 9×9 和 12×12 的特征维数,都仅为 LLE 算法的约 1/5,而且识别率最高。

表 3 的(a)和(b)比较了在保证最佳识别率的前提下,不同识别方法所需要的计算时间(整个实验运行在主频为 P4 2.4 GHz、内存为 512 MB、80 GB 硬盘、操作系统为 Windows XP sp2 的主机上面),其中,t_{E} 为特征提取时间,t_{C} 为分类时间,t_{T} 为总时间。在 Yale 人脸库中可以看出,PCA 在特征提取上耗费的时间最多,分别为 LLE 的 50 倍和本文方法的 37 倍。虽然 PCA 特征维数大,但是其分类时间小于后 2 种方法。同时,虽然 LLE 算法提取特征耗费的时间比 RLEBKF 短,但是由于其特征维数大,计算复杂度高,分类耗费的时间为本文提出的方法的 1.8 倍,所以总时间还是比本文提出的方法多。同样,在 AT&T 人脸库中也有类似结果。

表 2 3 种识别方法的最佳识别率和相应特征维数的比较

Tab. 2 Comparison of maximal recognition rate and corresponding feature dimension of three recognition methods

识别方法	$R_{\max}$(%)		D_{im}		“留一法” $R_{\max}$(%)	
	Yale	AT&T	Yale	AT&T	Yale (160/165)	AT&T (398/400)
PCA	65.0	74.2	100	150	86.2	89.3
LLE 算法	75.0	86.5	42×10	54×14	92.2	96.5
本文方法	93.5	95.6	9×9	12×12	98.75	99.5

表 3 在最佳识别率下,3 种识别方法在不同人脸库中的计算时间

Tab. 3 Comparison of CPU time of three recognition methods under maximal recognition rate on face database

单位:s

	识别方法	t_{E}	t_{C}	t_{T}
Yale	PCA	46.35	0.19	46.65
	LLE 算法	0.91	4.78	5.69
	RLEBKF	1.24	2.71	3.95
AT&T	PCA	54.25	0.68	54.93
	LLE 算法	2.28	6.54	8.82
	本文方法	3.68	5.35	9.03

5 结 论

LLE 算法非线性局部嵌入方法在生物数据分类、文字识别、脸谱图像处理中获得了一定的效果。但其存在固有的缺点,即该算法要求样本在流形上是稠密采样,对样本中的噪声很敏感,并且不能产生便于应用的、可延展的变换函数。针对其缺点,在 LLE 流形算法中引入核函数距离测度,以克服算法中由于高维稀疏空间所造成欧氏距离公式的点与点之间距离的对比性差等缺点,并从分析噪声对数据

集局部特性的影响入手,通过修正权值矩阵 $\boldsymbol{W}$,可以很好地减弱噪声对降维过程的影响;然后在损失函数中引入一个线性变换矩阵,利用非线性核技巧在维数很高的空间里求解子空间,最终达到最佳降维效果。理论分析及在 Yale 人脸库以及 AT&T 人脸库的测试结果表明,基于核函数的稳健线性嵌入方法性能优于 LLE 算法,和其他子空间方法相比,也具有更好的识别能力。

参考文献(References)

1 Kirby M, Sirovich L. Application of the karhunen2loeve procedure for the charaterization of human faces[J]. IEEE Transactions on Pattern A Nalysis and Machi ne Intelligence, 1990, **12**(1): 103-108.

2 Turk M, Pentland A. Eigenfaces for recognition[J]. Journal of Cognitive Neuroscience, 1991, **3**(1): 72-86.

3 Phillips P J, Moon H, Rizvi S, *et al*. The feret evaluation methodology for face recognition algorithms[J]. IEEE Transactions on Pattern A Nalysis and Machi ne Intelligence, 2000, **22**(10): 1090-1104.

4 Turk M, Pentalnd A. Eigenfaces for recognition[J]. Journal of Cognitive Neuroscience, 1991, **3**(1): 71-86.

5 Zhao W, Chellappa R, Rosenfeld A, *et al*. Face recognition: A literature survey [R]. CS-Tech Report4167, University of Maryland, 2000.

6 Roweis S T, Saul L K. Nonlmensionality reduction by locally linear embedding [J]. Science, 2000, **290**(22): 2323-2326.

7 Xu Rong, Jiang Feng, Yao Hong-xun. Overview of manifold learning [J]. Caai Transactions on Intelligent Systems, 2006, **1**(1): 44-51. [徐蓉, 姜峰, 姚鸿勋. 流形学习概述[J]. 智能系统学报, 2006, **1**(1): 44-51.]

8 Zhao Lian-wei, Luo Si-wei, Zhao Yan-chang, *et al*. Study on the low-dimensional embedding and the embedding dimensionality of manifold of high-dimensional data[J]. Journal of Software, 2005, **16**(8): 1423-1430. [赵连伟, 罗四维, 赵艳敞等. 高维数据的低维嵌入及嵌入维数研究[J]. 软件学报, 2005, **16**(8): 1423-1430.]

9 Bengio Y, Palement J, Vincent P, *et al*. Out-of-sample extensions for LLE, isomap, MDS, eigenmaps, and special clustering[A]. Neural Information Processing Systems [C], Cambridge, MA: M IT Press, 2003: 1238-1248.

10 Tenenbaum J B, De Silva V, Lan Gford J C. A global geometric framework for nonlinear dimensionality reduction [J]. Science, 2000, **290**(5500): 2319-2323.

11 Shashua A, Levin A, Avidan S. Manifold pursuit: A new approach to appearance based recognition [A]. In: Proceeding of 16th International Conference on Pattern Recognition[C], Quebec City, Canada, 2002: 590-594.

12 Weng Shi-feng, Zhang Chang-shui, Zhang Xue-gong. Nonlinear dimensionality reduction in the analysis of high dimensional medical data[J]. Journal of Tsinghua University Sci&Tech, 2004, **44**(4): 485-488.