doi:10.3969/j.issn.1001-4616.2016.01.002

一种基于迭代分解的增量流形学习算法

谈 超,吉根林

(南京师范大学计算机科学与技术学院,江苏 南京 210023)

[摘要] 流形学习可以用于发现大型高维数据集的内在结构,并给出理解该数据集的潜在方式,已被视为一种有效的非线性降维方法.近年来,新数据点不断地从数据流中产生,将改变已有数据点及其邻域点的坐标,传统流形学习算法不能有效地用于寻找高维数据流的内在信息.为了解决该问题,本文提出了一种基于迭代分解的增量流形学习算法 IMLID(Incremental Manifold Learning Algorithm Based on Iterative Decomposition),可以检测到数据流形中的逐步变化,校准逐渐变化中的流形,可提高在取样于真实世界的特征集上分类效果的精确率,利用真实数据集进行实验验证,结果表明本文提出的算法是有效的,与其他相关算法相比,其性能具有优势,在模式识别、生物信息等领域具有应用价值.

[关键词] 流形学习,迭代分解,增量流形学习

[中图分类号]TP181 [文献标志码]A [文章编号]1001-4616(2016)01-0014-07

An Incremental Manifold Learning Algorithm Based on Iterative Decomposition

Tan Chao,Ji Genlin

(School of Computer Science and Technology,Nanjing Normal University,Nanjing 210023,China)

Abstract: Manifold learning is used to discover intrinsic low-dimensional manifolds of data points embedded in high-dimensional spaces,which is useful in nonlinear dimension reduction. In recent years,new data points come continually, which will change the existing data points´ neighborhoods and their local distributions. Tranditional methods cannot discover intrinsic information of high dimensional data streams effectively. To solve this problem,we propose an Incremental Manifold Learning Algorithm Based on Iterative Decomposition(IMLID),which can detect the change of manifold and improve the classification accuracy of the feature set sampling in the real world. Experiments on real-life datasets validate the effectiveness of the proposed method which has important significance and extensive application value in pattern recognition and so on.

Key words: manifold learning,iterative decomposition,incremental learning

云计算[1]、物联网[2]、移动互联[3]、社交网络[4]等新兴信息技术和应用模式的快速发展,促使全球数据总量急剧增加,推动人类社会迈入大数据时代[5].流式大数据[6]作为大数据的一种重要形态,在商务智能、公共服务等诸多领域有着广泛的应用前景,并已在互联网、物联网等场景的应用中取得了显著的成效.

但流式大数据呈现出的实时性、无序性及突发性等显著特征,使得其与传统批量大数据在数据处理的要求、方式等方面有着明显的不同,也使得当前诸多模式识别和机器学习等相关算法无法直接应用到流式大数据处理中.在此背景下对流式大数据的分析和挖掘成为当前的热点研究课题[7,8].

从算法的层面来看,流式大数据的机器学习和分析挖掘问题,主要存在以下挑战:数据规模巨大、复杂性高,体现在数据对象数量及维度上,导致学习精度下降严重.与此同时,数据产生的规律性会随着时间变化而动态改变,这就对机器学习和分析挖掘算法的普适性(或泛化能力)提出了更高要求,导致目前已有很多相关算法失去了效力.

收稿日期:2015-09-20.

基金项目:江苏省高校自然科学基金(15KJB520022)、国家自然科学基金(41471371).

通讯联系人:谈超,博士,讲师,研究方向:机器学习、模式识别.E-mail:73022@njnu.edu.cn

近年来,许多流形学习算法被提出并应用于模式识别、数据挖掘和生物信息等领域.流形学习中的降维算法可以用于高维数据可视化.例如,1 组人脸图像在不同视角和光线环境下具有高分辨率,每个图像可以被 1 个数据点表示,具有高维空间中的像素值.由这些人脸图像构成的流形的内在维度被要求满足人脸识别的标准,小于实际图像大小.这些面部图像的内在结构信息可用流形学习算法进行降维并在低维空间中表示出来.因此,流形学习算法的 1 个目标在于根据嵌入在高维空间中的样本点构造非线性低维流形,通过将高维数据集转换到低维空间,可以保持数据集的内在结构并解决维数灾难等问题.

绝大多数流形学习算法都是在批量模式下进行,目的在于学习固定的数据集.然而,这些算法并不适合于动态数据集中的应用.对于动态数据集,原始数据集新输入数据以后重新运行整个算法十分耗时和低效,对数据集的存储和更新也是如此;同时,数据积累在流形学习中带来很大影响.

为了解决这些问题,本文提出了 1 种新的流形学习算法.首先,基于迭代更新系统,使用 1 种有效的算法来评价数据集更新的增量表达;接下来,采用增量的方法来学习数据流;最后,为了描述特征系统的动态特征,提出了 1 种新的增量更新过程.根据该方法,以前的样本可以舍去,当新样本加入时,无需存储整个矩阵.实验结果表明,提出算法的映射结果与批量算法相比,特别是当邻域大小增加时,更加精确,便于识别.

1　相关研究工作

这一节介绍相关研究工作:传统(非增量)流形学习算法和增量流形学习算法.

1.1　传统流形学习算法

当前大多数流形学习算法都是批处理模式,意味着所有数据点应被预先提供.等距映射算法 ISOMAP(ISOmetric MAPping)[9]试图在高维流形中保持测地距离来实现维数约减.局部线性嵌入映射算法 LLE(Locally Linear Embedding)[10]通过在每个数据点的邻域建立最优线性重构,将数据点嵌入到低维空间,尽可能好地保持局部几何特性.Laplacian 特征映射算法 LE(Laplacian Eigenmaps)[11]使用 Laplacian 图,研究嵌入顶点非线性映射问题,构造了加权矩阵并得到降维结果.Hessian 映射算法 HLLE(Hessian eigenmaps)[12]结合了 LLE 算法和 Hessian 映射的优势.局部切空间调准算法 LTSA(Line Tangent Space Alignment Algorithm)[13]为每个数据点构造了一个局部切空间,通过局部切空间的仿射变换获得数据点的全局低维嵌入坐标.

以批量模式处理数据是这些方法的共同特征,尽管它们已经被广泛地应用在许多领域,并取得了广泛的成功,但缺乏增量学习的能力.当数据点依次到达时,批处理的计算方式需要进行反复的学习,因为计算复杂度高使其不适合于增量学习.许多实际应用,例如数据流挖掘、视频监控和语音模式识别要求高维数据的实时嵌入.当新点加入时,批处理模式的方法通常在所有样本点上重复进行嵌入操作.因为大部分数据嵌入算法的时间复杂性为 $O(n^2)$,n 是数据点数目,这些算法的时间复杂度过高.故这些实际应用需要增量的流形学习算法,能够连续有效地更新基于新来的数据和现有数据的流形,无需在整个数据集上进行重复计算.

1.2　增量流形学习算法

近几年出现的增量流形学习算法,可以通过“遗忘”效应抛弃现有的重复信息,以适应对动态数据集的处理.增量学习算法的另一个优势在于对发展演变中的流形可视化,当越来越多的数据点到达时,流形的可视化能揭示数据流的一些变化特征.大多数现有的流形学习算法是针对 1 个固定的数据集进行学习.然而,数据集在实际应用中随着时间变化动态地改变.为了处理动态数据集的学习,增量流形学习算法将新数据点映射到低维空间,同时更新低维空间中已有数据点的坐标[14].

例如,Laplacian 特征映射的增量学习算法(Incremental Laplacianeigenmaps)[15]是通过保持局部邻域信息并使用子流形分析的方法,对数据集进行低维表达.该算法实际是在传统的批处理模式下进行.它引入了 1 个局部线性重建机制来添加新的近邻信息,并修正已有样本的低维嵌入结果.涉及的子流形方法需要解决$(k+1)\times(k+1)$的特征向量问题,总的时间复杂度 $O((k+1)^3)$ 较高.

Kouropteva 等人提出了增量局部线性嵌入算法 ILLE(Incremental Locally Linear Embedding)[16],估计

新样本的映射结果并重新计算原样本的映射值.代价矩阵建立在两点之间的局部距离上,当新数据点到达时保持不变,增量学习的问题由最小化函数来解决.特征问题通过最小化以下函数来获取新的投影点的坐标:$\min_{\boldsymbol{Y}_{\text{new}}}(\boldsymbol{Y}_{\text{new}}\boldsymbol{M}_{\text{new}}\boldsymbol{Y}_{\text{new}}^{T}-\text{diag}\{\lambda_1,\lambda_2,\cdots,\lambda_d\})$.

Abdel-Mannan 等在增量 LLE 算法[16]和增量 Hessian 局部线性嵌入算法 IHLLE(Incremental Hessian Locally Linear Embedding)[17]基础上提出了局部切空间校准算法的增量版本 ILTSA(Incremental Line Tangent Space Alignment Algorithm)[18].通过最小化重构误差获得新数据的低维嵌入坐标.类似于前面两个增量算法,ILTSA 的核心思想是利用两个增量更新的过程来适应新引入的数据点.第一个过程利用来自于数据点邻域的变换矩阵来寻找新的映射.在第二个过程中,新的投影点可以通过新加入点的 k 近邻计算校准矩阵来找到.这些都是用于形成属于 1 个流形邻域中的点的高维和低维坐标之间的关系.当 1 个新的数据点加入到高维流形时,ILTSA 的目的是适应该点并产生相应的校准矩阵.全局坐标和 LTSA 校准的方式一致,然而在整个数据集上计算成本较高.

Liu 等人提出了另 1 种增量 LTSA 算法[14].该方法修改了 LTSA 算法,通过利用 Ritz 加速进行子空间迭代,解决了矩阵规模增加的增量特征分解问题.该方法的问题类似于上文提到的 ILTSA 算法,即校准矩阵需要重建来包括新加入的点,当数据集规模非常大时不是很实用.

总之,ILLE 直接通过评估协方差矩阵的特征值和特征向量来解决该问题,得到新点的低维映射坐标.ILE 和 ILTSA 根据局部几何信息估计全局坐标,依照每个新点和其近邻点之间的关系,实际上是属于利用邻域关系的嵌入方法.

现有的增量流形学习算法具有以下缺陷:对于一些现有的增量学习算法,增加新的数据点可能改变当前邻域和流形的局部分布,新样本的加入导致删除图中的临界边,随后大幅度地改变测地距离,将会发生短回路或空洞现象,例如增量 Isomap 算法 IISOMAP(Incremental ISOmetric MAPping)[19].计算成本的负担是另一个限制,因为迭代优化问题需要计算的部分未知,以前的工作主要是采用迭代算法计算整个过程,计算量很大,例如增量 LTSA 算法.现有增量方法的另一限制在于近似误差没有保障.例如,在增量 PCA 算法 IPCA(Incremental PCA)[20,21]中,新样本依次加入,从原始训练图像和新添加的样本重构特征空间,通过使用原始图像的低维系数向量获得更新后的特征空间.所以这些方法受到不可预测近似误差的影响,同样的问题也存在于其他增量方法.

从以上相关工作的综述可以看出,现有的算法不适合在动态数据集上的应用,因此,本文基于增量方式,提出了 1 种新的流形学习算法.

2　一种基于迭代分解的增量流形学习算法

我们知道 Laplacian 映射算法通过合并数据集的邻域信息建立 Laplacian 图,并通过最优保持局部邻域信息计算数据集的低维表示.具体地说,它构造了 1 个 Laplacian 图,带有连接邻域点的边.通过计算 Laplacian 图的特征向量获得嵌入映射,可以保持 1 个数据集低维表示的局部邻域.在这里,我们提出 1 种更有效的算法,基于迭代生成的特征值系统,增量地表示 1 个更新的数据集.

降维的一般问题可以表示如下:给出 1 组 n 个点 $\boldsymbol{X}=[x_1,x_2,\cdots,x_n]$ $(x_i\in\mathbf{R}^l)$,找出一组 $y_1,y_2,\cdots,y_n(y_i\in\mathbf{R}^m,\mathrm{m}\ll\mathrm{l})$ 使得 y_i 尽可能精确的表示 x_i.我们给出一个带权重的图:$G=G(V,\boldsymbol{W})$,V 是顶点集,$\boldsymbol{W}$ 是权重.在本文中,假设入射矩阵 $\boldsymbol{W}$ 是权值矩阵,$w_{ij}=1/k$ 当点 i 和点 j 互为 k 近邻,否则 $w_{ij}=0$. 对角矩阵 $\boldsymbol{D}$ 定义为:$\boldsymbol{D}=\text{diag}\{d_1,d_2,\cdots,d_n\}$,$d_i=\sum_j w_{ij}$.算法中主要使用的符号说明如表 1 所示.

表 1　算法主要用到的符号说明

Table 1　The main symbols used in the algorithm

符号	说明
$\boldsymbol{X}$	数据矩阵 $\boldsymbol{X}=[x_1,x_2,\cdots,x_n]$
u_i	向量 $\boldsymbol{u}$ 的第 i 个元素
$\boldsymbol{R}_{ij}$	$\boldsymbol{R}$ 中下标为(i,j)的元素
$\|\cdot\|$	表示 l_2 范数,即 $\|\boldsymbol{x}\|=\sqrt{\boldsymbol{x}^T\boldsymbol{x}}$
$\boldsymbol{L}$	拉普拉斯矩阵

2.1　迭代更新

首先,我们考虑一种迭代过程,在低维特征空间中构造新点的嵌入坐标,这在我们的工作[22]中提到

过.根据 Laplacian 矩阵 $\boldsymbol{L}$ 的定义,我们知道 $\{\boldsymbol{L}_n\},\|\boldsymbol{L}_n\|<\infty$ 及 $\lim\limits_{k\to\infty}\boldsymbol{L}_k=\boldsymbol{L}$.故我们有 $\lim\limits_{n\to\infty}\frac{1}{n}\sum\limits_{k=1}^{n}\boldsymbol{L}_k=\boldsymbol{L}$.将 $\boldsymbol{L}_k$ 的特征向量表示为 $\boldsymbol{v}_k$,λ_k 作为特征值,即 $\boldsymbol{L}_k\boldsymbol{v}_k=\lambda_k\boldsymbol{v}_k$.定义 $\boldsymbol{u}_k=\boldsymbol{L}_k\boldsymbol{v}_k$,因此 $\lim\limits_{k\to\infty}\boldsymbol{u}_k=\boldsymbol{L}\boldsymbol{v}=\lambda\boldsymbol{v}$.设 $\boldsymbol{M}_n=\frac{1}{n}\sum\limits_{k=1}^{n}\boldsymbol{u}_k$,$\boldsymbol{M}_n$ 可被写为增量估计的递归形式,如下所示:

$$\boldsymbol{M}_n=\frac{1}{n}\sum_{k=1}^{n}\boldsymbol{u}_k=\frac{1}{n}\sum_{k=1}^{n}\boldsymbol{L}_k\boldsymbol{v}_k=\frac{n-1}{n}\left(\frac{1}{n-1}\sum_{k=1}^{n-1}\boldsymbol{L}_k\boldsymbol{v}_k\right)+\frac{1}{n}\boldsymbol{L}_n\boldsymbol{v}_n=\frac{n-1}{n}\boldsymbol{M}_{n-1}+\frac{1}{n}\boldsymbol{L}_n\boldsymbol{v}_n. \tag{1}$$

由于 $\boldsymbol{v}_n$ 是 $\boldsymbol{L}_n$ 的特征向量,可通过 $\frac{\boldsymbol{M}_{n-1}}{\|\boldsymbol{M}_{n-1}\|}$ 来估计,我们有 $\lim\limits_{n\to\infty}\boldsymbol{v}_n=\boldsymbol{v}\approx\lim\limits_{n\to\infty}\frac{\boldsymbol{M}_{n-1}}{\|\boldsymbol{M}_{n-1}\|}$,故:

$$\boldsymbol{M}_n\approx\frac{n-1}{n}\boldsymbol{M}_{n-1}+\frac{1}{n}\boldsymbol{L}_n\frac{\boldsymbol{M}_{n-1}}{\|\boldsymbol{M}_{n-1}\|}. \tag{2}$$

命题 1　给出一个 $n\times n$ 的 Laplacian 矩阵 $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{W}$,存在一个关联矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{R}$ 使得 $\boldsymbol{L}=\boldsymbol{W}\boldsymbol{R}\boldsymbol{R}^{\mathrm{T}}$,$\boldsymbol{R}$ 包含所有列向量 $\boldsymbol{r}_{ij}(1\leqslant i<j\leqslant n)$[23].

基于以上分析及命题 1,我们通过在式(2)中 $\boldsymbol{L}_n$ 的位置插入 $\boldsymbol{L}=\boldsymbol{W}\boldsymbol{R}\boldsymbol{R}^{\mathrm{T}}$ 得到 $\boldsymbol{M}_n$ 的迭代表示:

$$\boldsymbol{M}_n\approx\frac{n-1}{n}\boldsymbol{M}_{n-1}+\frac{1}{n}\boldsymbol{W}_n\boldsymbol{R}_n\boldsymbol{R}_n^{\mathrm{T}}\frac{\boldsymbol{M}_{n-1}}{\|\boldsymbol{M}_{n-1}\|}. \tag{3}$$

因为 $\boldsymbol{M}_1=\boldsymbol{u}_1=\boldsymbol{L}_1\boldsymbol{v}_1$,初始的 $\boldsymbol{M}$ 可通过式(3)迭代地计算.因为 $\boldsymbol{L}$ 是矩阵,且 $\boldsymbol{v}$ 是向量,故 $\boldsymbol{u}=\boldsymbol{L}\boldsymbol{v}$ 属于向量,我们将上述迭代过程中得到的向量 $\boldsymbol{u}_i$ 存入向量矩阵 $\boldsymbol{U}$,$\boldsymbol{U}=[\boldsymbol{u}_1,\cdots,\boldsymbol{u}_n]$.矩阵 $\boldsymbol{U}$ 由包括其自身的 k 近邻组成(欧式距离意义上).为了构造 $\boldsymbol{U}$,我们提出 1 种新的基于迭代分解的增量流形学习算法,来计算新点包括已有点在低维空间中的嵌入坐标.

2.2　增量学习

给出上文中得到的矩阵 $\boldsymbol{U}$ 中的向量 $\boldsymbol{u}_i$,每当新样本到达时存储并更新整个矩阵十分低效.为了解决这个问题,我们提出了一种增量更新过程,在该过程中我们只需存储平均向量 $\bar{\boldsymbol{M}}_n$ 和矩阵 $\hat{\boldsymbol{G}}_{\mathrm{new}}$,这样老的样本点都可丢弃.

基于式(3)和 M_n 的定义:$\boldsymbol{M}_n=\frac{1}{n}\sum\limits_{k=1}^{n}\boldsymbol{u}_k$,令 $\bar{\boldsymbol{M}}_n$ 为现有 n 个 $\boldsymbol{u}_k$ 的平均值.我们将其对于 $\boldsymbol{u}_{\mathrm{new}}$ 的更新形式设置为:

$$\bar{\boldsymbol{M}}_{\mathrm{new}}=\frac{n-1}{n}\bar{\boldsymbol{M}}_n+\frac{1}{n}\boldsymbol{u}_{\mathrm{new}}. \tag{4}$$

这提示我们基于当前样本和之前均值的基础上得到新的均值的递归形式.令 $\boldsymbol{K}_i=\left(\boldsymbol{I}-\frac{1}{k}\boldsymbol{e}_i\boldsymbol{e}_i^{\mathrm{T}}\right)(\boldsymbol{I}-\boldsymbol{R}_i\boldsymbol{R}_i^{\mathrm{T}})$,我们知道矩阵 $\boldsymbol{K}$ 是在由 $\boldsymbol{R}$ 的列向量张成的子空间上的投影.故 $\boldsymbol{K}_i$ 中的 $\boldsymbol{R}_i\boldsymbol{R}_i^{\mathrm{T}}$ 可以由计算 $\boldsymbol{U}\left(\boldsymbol{I}-\frac{1}{k}\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)$ 的特征向量而得到.

基于向量 $\boldsymbol{U}\left(\boldsymbol{I}-\frac{1}{k}\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)$ 和等式(4),我们对矩阵 $\boldsymbol{G}=\left[\boldsymbol{U}\left(\boldsymbol{I}-\frac{1}{k}\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)\right]^{\mathrm{T}}\boldsymbol{U}\left(\boldsymbol{I}-\frac{1}{k}\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)$ 的更新形式可以由下式估计:

$$\hat{\boldsymbol{G}}_{\mathrm{new}}=\frac{n-1}{n}\hat{\boldsymbol{G}}_n+\frac{1}{n}\left(\boldsymbol{u}_{\mathrm{new}}-\bar{\boldsymbol{M}}_{\mathrm{new}}\right)^{\mathrm{T}}(\boldsymbol{u}_{\mathrm{new}}-\bar{\boldsymbol{M}}_{\mathrm{new}}). \tag{5}$$

为了增量地更新所有点在低维空间中的嵌入坐标,我们将用如下方式更新矩阵.

首先,我们对 $\hat{\boldsymbol{G}}_{\mathrm{new}}$ 做特征分解,并获得它的前 d 个特征向量 $\boldsymbol{g}_i$ 和特征值 λ_i,$i=1,2,\cdots,d$.接下来我们基于 $\boldsymbol{g}_i$ 和 λ_i 构造 d 个向量:$\boldsymbol{a}_i=\boldsymbol{g}_i\sqrt{\frac{n-1}{n}\lambda_i}$.而 $\boldsymbol{a}_{d+1}=\sqrt{\frac{1}{n}}(\boldsymbol{u}_{\mathrm{new}}-\bar{\boldsymbol{M}}_{\mathrm{new}})^{\mathrm{T}}$ 对应于新加入的点.那么包含上述向量的矩阵可以定义为:$\boldsymbol{A}=[a_1,a_2,\cdots,a_d,a_{d+1}]$.

接下来,可用式 $\boldsymbol{B}=\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ 表达一个内积矩阵,故矩阵 $\boldsymbol{B}$ 是 $\boldsymbol{A}$ 的列向量张成的子空间上的正交投影.上文定义的 $\boldsymbol{K}_i$ 中的矩阵 $\boldsymbol{R}_i\boldsymbol{R}_i^{\mathrm{T}}$ 是由 $\boldsymbol{R}$ 的列向量张成的子空间上的投影,可通过计算 $\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}}$ 的 d 个最大特征值对应的特征向量来获得.根据 $\boldsymbol{B}$ 的定义,$\boldsymbol{r}_1,\cdots,\boldsymbol{r}_d$ 可通过计算 $\boldsymbol{B}$ 的前 d 个最小奇异向量得到[24].

2.3　算法步骤

基于上文讨论的迭代方式,我们提出1种新的流形学习算法称为基于迭代分解的增量流形学习算法IMLID. 该算法首先构造了数据集加入新样本点后的迭代更新结构,考虑增量思想,结合前面的迭代更新结构,算法IMLID可以表示为表2所示.

表2　算法IMLID的主要步骤

Table 2　The main steps of the algorithm IMLID

算法:IMLID
1. 将 $\boldsymbol{Ly}=\lambda\boldsymbol{Dy}$ 作为标准Laplacian特征分解,得到最小特征值对应的特征向量,结果作为初始 $\boldsymbol{v}_1$ 和 $\boldsymbol{L}_1$. 2. 因为 $\boldsymbol{M}_1=\boldsymbol{u}_1=\boldsymbol{L}_1\boldsymbol{v}_1$,用式(3)更新 $\boldsymbol{M}_n$. 将上述迭代过程中得到的向量 $\boldsymbol{u}_i$ 存入向量矩阵 $\boldsymbol{U}$, $\boldsymbol{U}=[\boldsymbol{u}_1,\cdots,\boldsymbol{u}_n]$. 3. 计算已有 $\boldsymbol{u}_k$ 的平均值 $\bar{\boldsymbol{M}}_n$,当一个新的数据点加入时,用式(4)更新 $\bar{\boldsymbol{M}}_{\text{new}}$. 4. 基于第二步中得到的向量矩阵 $\boldsymbol{U}$,得到矩阵 $\boldsymbol{G}=\left[\boldsymbol{U}\left(\left(\boldsymbol{I}-\frac{1}{k}\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)\right)\right]^{\mathrm{T}}\boldsymbol{U}\left(\boldsymbol{I}-\frac{1}{k}\boldsymbol{e}\boldsymbol{e}^{\mathrm{T}}\right)$,$\boldsymbol{G}$ 可以用式(5)进行更新. 5. 对 $\hat{\boldsymbol{G}}_{new}$ 进行特征分解,基于得到的特征向量构造矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$. 计算 $\boldsymbol{B}$ 的 d 个最小奇异向量,得到:$\boldsymbol{r}_1,\cdots,\boldsymbol{r}_d$. 根据 $\boldsymbol{E}_i=[e/\sqrt{k},\boldsymbol{r}_1,\cdots,\boldsymbol{r}_d]$,可以计算得到 $\boldsymbol{L}_i=\boldsymbol{W}_i\boldsymbol{R}_i\boldsymbol{R}_i^{\mathrm{T}}$. 故对更新以后的 $\boldsymbol{L}$ 计算最小特征值对应的特征向量,构成算法的结果.

3　算法复杂度分析

算法IMLID的第一步需要解一个 $k\times k$ 的特征问题,k 是近邻点数目,解该 $k\times k$ 特征问题的时间复杂度为 $O(k^2)$.

对于后续的步骤,计算复杂度集中在式(4)和(5),计算量最大的步骤集中于高维空间数据的乘积. 对于式(4)一共需要做 $k(k+1)/2$ 次点积. 式(5)中每一步估计有一个额外的乘积 $(\boldsymbol{u}_{\text{new}}-\bar{\boldsymbol{M}}_{\text{new}})^{\mathrm{T}}(\boldsymbol{u}_{\text{new}}-\bar{\boldsymbol{M}}_{\text{new}})$. 式(5)的每个迭代步骤中节省的计算复杂度为 $(k-1)/2$,减去之后,经过 n 次迭代步骤后的平均计算复杂度为 $O(nk^2)$. 除了以上步骤,算法IMLID的计算复杂度主要是线性的,包括行列式的乘积及一些简单的加法. 所以核心计算复杂度集中于以上几个步骤,故算法IMLID计算的最终复杂度为 $O(nk^2)$.

4　实验结果与分析

我们通过一系列实验来分析算法IMLID的性能和优势,并与其他相关算法如增量局部线性嵌入(ILLE)、增量局部切空间校准(ILTSA)、增量拉普拉斯映射(ILE)、增量海森映射(IHLLE)和增量主成分分析(IPCA)等算法进行比较. 算法IMLID用Matlab实现,实验环境为一台1.8 GHz CPU,4GB内存的PC. 实验所用数据取自一些典型的非线性流形学习数据集如UCI数据集,及来自真实面部图像如Yale B人脸库.

4.1　UCI实验验证与结果分析

我们用UCI数据集中的不同数据集进行降实验,并测试各种增量算法的识别率. UCI数据集中的5种数据集(Iris,Wine,Glass,Movement,Cloud)的基本信息如表3所示.

表3　UCI数据集的相关信息

Table 3　Information of several UCI datasets

UCI数据集	样本点数	特征数	分类数
Iris	150	4	3
Wine	178	13	3
Glass	214	10	6
Movement	360	91	15
Cloud	1 024	10	8

我们首先用不同算法将Iris数据集降到二维以后,再用k-NN分类法在该数据集上进行分类. Iris数据集包括150个样本,具有4种特征,从中选择60个样本作为训练集,剩下的90个作为测试集. 将低维

数据集分为 3 类以后,与数据集中的正确分类进行比较,来计算识别率 . 重复实验 10 次以后,各种增量学习算法在 Iris 数据集上的平均识别率如表 4 所示 . 同样的,我们在其他 4 种 UCI 数据集上进行降维以后的分类识别率结果也表示在表 4 中 . 从表中我们可以看出,算法 IMLID 与 ILLE、IHLLE 和 IPCA 相比优势明显,相对于其他算法来说取得了不错的平均识别率 .

表 4　不同算法在 UCI 数据集上的平均识别率

Table 4　The average recognition rate of different algorithms on UCI datasets

	ILLE	ILTSA	ILE	IHLLE	IPCA	IMLID
Iris	0.726 4	0.918 4	0.957 1	0.671 3	0.773 8	0.922 76
Wine	0.783 7	0.916 6	0.927 9	0.634 1	0.683 3	0.933 63
Glass	0.758 5	0.947 2	0.935 4	0.696 6	0.742 4	0.949 72
Movement	0.716 9	0.943 5	0.945 2	0.620 6	0.746 4	0.963 33
Cloud	0.886 4	0.991 5	0.973 2	0.676 3	0.763 1	0.994 50

4.2　Yale B 实验验证与结果分析

Yale 人脸库由 15 个不同的个体,每个个体 11 张图像,共 165 张正面人脸图像组成,每个面部表情图像样本的原始像素值为 32×32,该数据集包含了不同人脸面部表情、人脸姿态变化以及不同的光照条件,图 1 是该库的人脸图像示例 .

图 1　Yale 人脸库图像

Fig.1　Face image of Yale face dataset

实验采用不同增量学习算法将样本的维数分别降至 10 维、20 维直到 150 维,然后在降维后的 Yale 数据集中任意选取 1 幅作为训练集,剩余的图像作为测试集,并计算分类精确度,图 2 展示了不同增量算法将 Yale 数据集降到不同维数的分类结果 . 由图中可知,本文提出的算法 IMLID 将人脸数据降维以后的分类精确度在大多数范围内优于其他算法,特别是在降到 20 维到 120 维这个区间,具有明显的优势,可以看出算法 IMLID 对后续分类学习起到了积极作用,在很多应用比如模式识别、图像处理等具有应用价值 .

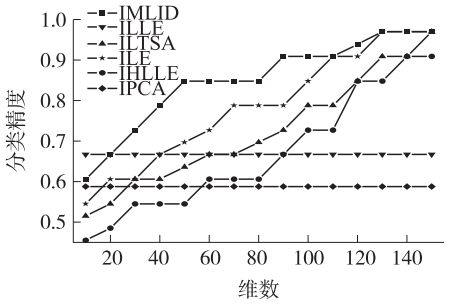

图 2　不同算法在 Yale 人脸数据集上的分类精度

Fig.2　The classification accuracies of different algorithms on Yale face dataset

5　结论与展望

本文提出了 1 种基于迭代分解的增量流形学习算法,用迭代的方式计算关系矩阵的特征值和特征向量,从而获得数据集投影到低维空间的嵌入坐标 . 具有以下特点:

1)由于具有非参数的特性,可以很好地检测到数据流形中的逐步变化,特别适合处理数据积累的问题;

2)可校准逐渐变化中的流形;

3)可提高在取样于真实世界的特征集上聚类效果的精确率,在人脸识别,生物信息等领域具有重要意义及广泛的应用价值 .

算法的不足之处是需要对矩阵进行运算,在对大量数据组成的矩阵求解时需要大量内存空间,这成为制约该算法性能的瓶颈 . 如何利用稀疏表示的方式对原始数据集提取有效特征,对数据矩阵进行压缩以提高运算效率是未来研究的方向 .

[参考文献]

[1] ARMBRUST M,FOX A,GRIFFITH R,et al. A view of cloud computing[J]. Communications of the ACM,2010,53(4):50-58.

[2] ATZORI L,IERA A,MORABITO G. The Internet of things:a survey[J]. Computer networks,2010,54(15):2 787-2 805.

[3] FUNG P T,LIN C,LI Z Z,et al. DragonNet:a robust mobile internet service system for long-distance trains[J]. IEEE transactions on mobile computing,2013,12(11):2 206-2 218.

[4] HAEWOON K,CHANGHYUN L,et al. What is twitter,a social network or a news media?[C]//Proceedings of the 19th International Conference on World Wide Web,Raleigh,2010:591-600.

[5] LABRINIDIS A,JAGADISH H V. Challenges and opportunities with big data[J]. Proc VLDB Endow,2012,5(12):2 032-2 033.

[6] XUE Q ZENG,GUO Z L. Incremental partial least squares analysis of big streaming data[J]. Pattern recognition,2014,47(11):3 726-3 735.

[7] GULISANO V. Streamcloud:an elastic and scalable data streaming system[J]. IEEE transactions on parallel and distributed systems,2012,23(12):2 351-2 365.

[8] LEI C,RUNDENSTELNER E. Robust distributed query processing for streaming data[J]. ACM transactions on database system,2014,39(2):1-45.

[9] TENENBAUM J B,SILVA D V,LANGFORD J C. A global geometric framework for nonlinear dimensionality reduction[J]. Science,2000,290:2 319-2 323.

[10] ROWEIS S T,SAUL L K. Nonlinear dimensionality reduction by locally linear embedding[J]. Science,2000,290:2 323-2 326.

[11] BELKIN M,NIYOGI P. Laplacian eigenmaps for dimensionality reduction and data representation[J]. Neural Computation,2003,15(6):1 373-1 396.

[12] DONOHO D L,GRIMES C. Hessian eigenmaps:locally linear embedding techniquesfor high-dimensional data[J]. Proceedings of the national academy of sciences,2003,100:5 591-5 596.

[13] ZHANG Z Y,ZHA H Y. Principal manifolds and nonlinear dimensionality reduction viatangent space alignment[J]. SIAM journal of scientific computing,2004,26:313-338.

[14] LIU X M,YIN J W,FENG Z L,et al. Incremental manifold learning via tangent space alignment[C]//Artificial Neural Networks in Pattern Recognition,Ulm,Germany,2006:107-121.

[15] JIA P,YIN J,et al. Incremental Laplacianeigenmaps by preserving adjacent information between data points[J]. Pattern recognition letters,2009,30:1 457-1 463.

[16] KOUROPTEVA O,OKUN O,et al. Incremental locally linear embedding[J]. Pattern recognition,2005,38:1 764-1 767.

[17] ABDEL M O,BEN H A,et al. Incremental Hessian locally linear embedding algorithm[C]//The 9th International Symposium on Signal Processing and Its Applications,Sharjah,United Arab Emirates,2007:1-4.

[18] ABDEL M O,BEN H A,et al. Incremental line tangent space alignment algorithm[C]//Canadian Conference on Electrical and Computer Engineering,Vancouver,BC,2007:1 329-1 332.

[19] LAW M H C,JAIN A K. Incremental nonlinear dimensionality reduction by manifold learning[J]. IEEE Trans on Pattern Analysis and Machine Intelligence,2006,28:377-391.

[20] LI Y M,XU L Q,MORPHETT J,et al. An integrated algorithm of incremental and robust PCA[C]//International Conference on Image Processing,Barcelona,2003(1):245-248.

[21] LI Y M. On incremental and robust subspace learning[J]. Pattern recognition,2004,37:1 509-1 518.

[22] TAN C,GUAN J H. A new manifold learning algorithm based on incremental spectral decomposition[C]//Advanced Data Mining and Applications,Nanjing,2012.

[23] NING H Z,XU W,et al. Incremental spectral clustering by efficiently updating the eigen-system[J]. Pattern recognition,2010,43:113-127.

[24] GOLUB G H,VAN L C F. Matrix computations[M]. Baltimore:Johns Hopkins University Press,2012.

[责任编辑:顾晓天]