一种高维数据类模板的设计方法与应用

肖化昆

(广东技术师范学院计算机科学系 广州 510665)

摘 要 本文构建了一种新的高维数据类模板。高维数据类模板是一个通用数据类型,其中封装了高维数据的数据结构和基本降维算法,能灵活描述和处理多种类型的高维数据对象,克服了传统方法的局限,具有可维护、可移植和可扩充性的特点。本文给出了定义高维数据类模板的部分 C++ 源代码及一个应用实例。

关键词 高维数据,类模板,数据结构

Design and Application about High Dimension Data Class Template

XIAO Hua-Kun

(Guangdong Polytechnic Normal University, Guangzhou 510665)

Abstract This paper constructs one kind of new class template which deal with high dimension data. The class template of high dimenston data is an general software framework, it has encapsulated the datastructures and algorithms, can nimbly describe and operate the many kinds types of high dimension data. Thisnew method based on parameterized class, can overcome the traditional method limitation, it has the characteristics such as maintainable, transplantable and extensible. This paper produced the kind of template partial C++ source code and a concrete application example.

Keywords High dimension data, Class template, Data structures

1 引言

自然界中存在大量复杂事物和现象,为了提供多方面、完整的信息,需要用多变量组成的向量数据表示,这些用多个变量描述观察对象的数据,抽象出来就是高维数据[2]。由于高维数据存在普遍性,使得对高维数据研究有着非常重要的意义,但高维数据自身表达和处理复杂,常常妨碍了它的实际应用。如何有效地分析处理大量的高维数据信息,用简便可行的方法从原始数据中提取有用信息,进而用可靠的数学模型来描述、推测观测数据,是当前研究的热点。一个可行的办法是利用 C++ 语言的模板技术,构造出一种新的抽象数据类型,建立一个高维数据类模板,该类模板是一个参数化类,能灵活描述和处理多种类型的高维数据对象并建立数学模型。

2 数据结构与算法

2.1 高维对象的数据结构

高维对象的信息经转化可以存储在一个二维数组中。例如,一个 N 维对象(N 个特征刻画的对象),可以用 N 维向量 $x=(x_1, x_2, \cdots, x_n)$ 表示,其中 x_j 表示该对象的第 j 个特征值。m 个 N 维对象的集合可用数组 $(x_{ij})i=1,\cdots,m,\ j=1,\cdots,n$ 表示,其中元素 x_{ij} 表示对象集合中第 i 个对象的第 j 个特征值。对于连续对象,如一条温度变化曲线,若能将时间区间划分成 N 等分,标记时刻 j 的温度值为 t_j 度,则向量 $t=(t_1, t_2, \cdots, t_n)$ 就能近似表示这条温度曲线,一组温度曲线也同样可以用二维数组 $(t_{ij})i=1,\cdots,m,\ j=1,\cdots,n$ 近似表示。

2.2 高维数据的降维算法

高维数据能充分表达复杂事物的信息,但随着维数的提高,算法复杂程度将急剧增大。因此,在高维数据研究中,"降维"是一项关键性技术。利用降维算法把高维数据通过某种组合投影到低维子空间上,寻找出能反映原高维数据结构或特征的投影,在低维空间上对数据结构进行简化和可视化处理,以达到分析研究高维数据的目的。将高维空间中的数据降低成低维空间中的数据,这个过程也叫特征提取。传统的降维方法可以分为线性降维方法和非线性降维方法两类。针对不同的目的,已有许多可行的降维方法[1~3]。本文介绍一种实用的非线性映射方法,本文作者应用此方法对影响洁酶素生长的数十种因素进行分析,找出了少数几个关键因素,对关键因素进行最优控制,收到较好效果[4]。

非线性映照方法[5]的原理是将高维空间的点集映射到二维空间,在映射中尽量保持各点间的距离结构不变,即维持平方和的意义下的各点距离变化最小。设上述 N 维空间中的点集 $X(x_1, x_2, \cdots, x_n)$,其第 i 个点的坐标为 $(x_{i1}, x_{i2}, x_{i3}, \cdots, x_{in})$,点 i,j 间距离为:

$$d_{ij}^{*}=\sqrt{\sum_{k=1}^{\infty}(x_{ik}-x_{jk})^2}$$

映射到二维空间后,第 i 点坐标为 (y_{i1}, y_{i2}),点 i,j 间距离为:$d_{ij}=\sqrt{\sum_{k=1}^{2}(y_{ik}-y_{jk})^2}$

则维持 $D_{ij}^{*}=D_{ij}$ 的方程为矛盾方程组,但可定义下述误差函数:$E=\frac{1}{\sum_{i<j}^{n}d_{ij}^{*}}\sum_{i<j}^{n}\frac{(d_{ij}^{*}-d_{ij})^2}{d_{ij}^{*}}$

使其极小化,以实现从高维空间到二维空间的"降维"。即将 N 维空间中的点 $X(x_1, x_2, \cdots, x_n)$ 映射为二维空间的点 $Y(y_1, y_2)$。本文基于该方法设计了一种函数模板 NLinerMap(T),其中的参数 T 是泛型参数,该函数模板可以将 N 维空间点的集合 $(X_{ij})_{m\times n}$ 映射到二维空间,变换结果为二维点集 $(Y_{ij})_{m\times 2}$。

其它一些有代表性的降维方法还有 Fisher、PCA、PLS、和 LLE 方法等。其中,由 Sam T. Roweis 和 Lawrence K. Saul 提出的 LLE(Locally linear embedding)算法,是一种新的针对非线性数据的降维方法,并且能够使降维后的数据保持原有的拓扑结构,已经广泛应用于图像数据的分类与聚类、文字识别、图像识别、数据可视化、以及生物和化学信息处理等领域中。LLE 算法可以归结为三步:①寻找每个样本点的 k

肖化昆 高级工程师,主要研究方向:软件工程、计算机数字仿真。

个近邻点；②由每个样本点的近邻点计算出该样本点的局部重建权值矩阵；③由该样本点的局部重建权值矩阵和其近邻点计算出该样本点的输出值。本文基于 LLE 算法设计的函数模板是 UltLee3_2(T)。

2.3 高维数据可视化

图形识别是人的一种本能。人对二维(平面)图形的识别能力很强，对三维(立体)图形也有识别能力，但不能识别四维以上空间的图形。在科学技术中，用作图法处理数据，找寻规律是很有效的。但对多因素影响的事物，人们通常采用"在高维空间中取一系列剖面"，通过一系列剖面图形来认识整体。但这种方法有很大局限性。使用高维数据图形识别技术，是解决上述问题的一种新途径。人们把高维数据"降维"后"绘"成二维或三维空间中的图形，凭借人的脑力就可直观地发现其规律性。本文设计的二维图形函数模板 Dot_Map(T)，可以显示多种高维数据降维后的二维图形。

3 类模板的构造与实现

类模板是高维数据结构及相关算法的封装体，其类型参数不是一种固定的数据类型，在实例化时编译器用实际的数据类型来代替它，因此类模板能适应不同类型的高维数据，可实现数据类型参数化处理。下面是用 C++ 语言编写的高维数据类模板和函数模板的一部分源代码。

```
template <class T>
class HDDataTempleClass
{
public:
  HDDataTempleClass():HighDimRow(0), HighDimCol(0){;}
  HDDataTempleClass(int hdRow, int hdCol)
  { HighDimRow = hdRow;
    HighDimCol = hdCol;
    for(int i=0; i < hdRow; i++)
    { vector<T> x(hdCol);
      int y = x.size();
      HDDataInstance.push_back(x);
    }
  }
  T GetAt(int hdRow, int hdCol)
  { if (hdRow >= HighDimRow ||hdCol >= HighDimCol)
      throw out_of_range("Array out of bound");
    else
      return HDDataInstance[hdRow][hdCol];
    }
    void GrowCol(int newSize)
    { if(newSize <= HighDimCol)return;
    HighDimCol = newSize;
  for(int i=0; i < HighDimRow; i++)
    HDDataInstance[i].resize(newSize);
  }
  vector<T>& operator[](int x)
  { return HDDataInstance[x]; }
private:
      vector<vector <T>> HDDataInstance;
      unsigned int HighDimRow;
      unsigned int HighDimCol;
};
```

用类模板创建高维数据对象的过程非常简单。只要在客户程序中执行语句：HDDataTempleClass <Double> HDDataInstance(m,n)，就能创建出了一个 $m \times n$ 个元素的高维数据对象，该高维对象的数据类型是 Double 型的。在类模板 HDDataTempleClass 中定义的数组是动态的，其行、列数都可利用类模板中定义的模板函数加以改变。此外，该高维数据类模板已经重载了高维数据计算所需的各种专用操作符，例如下标运算符"[ ]"等，因此，该高维数据类模板可以像常规 C++ 的二维数组一样使用"[ ]"等运算符。

4 应用实例

使用高维数据类模板创建和处理高维数据对象的过程如图 1 所示。

在研究洁酶素工艺参数控制中[4]，我们将洁酶素产出量随时间增长变化的过程(增长曲线)划分成初期、中期和后期三个时段，每一时段选取一些观测点记录其产出量，连同其它生化过程参数，构建了一个标有 17 个坐标的高维数据空间，采集日常生产中 180 个批次的记录数据，记入到数据文件 UserFile.dat 中，然后通过客户程序调用前述高维数据类模板创建出高维数据对象 HDDataInstance[180][17]，并通过该对象调用模板函数，依次进行降维计算和在二维平面上绘制"点"的分布图，图 2 是图形显示结果，可以明显看出两类产出结果的分布规律，其中"*"代表高产批次的"曲线"，"·"代表低产批次的"曲线"。以下是该客户程序的主函数的部分源代码。

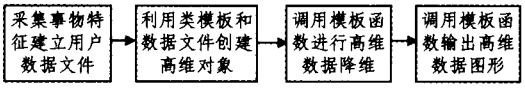

图 1 高维数据对象的处理过程

```
void main()
{ ……
  m=GetSampleFile(&UserFile); // 获取高维对象样本数目;
  n =GetFeatureFile(&UserFile); // 获取高维对象特征数目;
  HDDataTempleClass<double>HDDataInstance(m,n);// 创建高维数据结构;
  HDDataInstance.ReadUserFile(&UserFile); // 导入数据,创建高维实例;
  HDDataInstance. NLinerMap(); //对高维数据实施降维计算;
  HDDataInstance.D2_Dot_Map(); //在二维平面中绘制散点分布图;
  ……
}
```

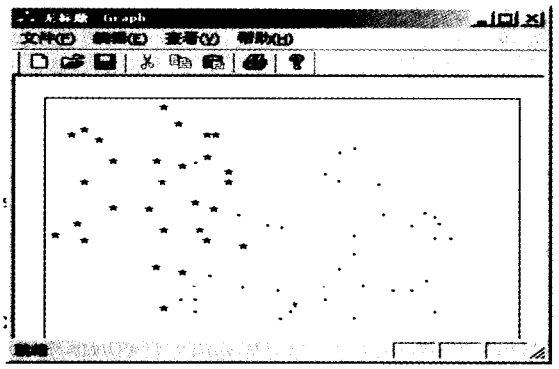

图 2 具有 17 个特征的洁酶素发酵过程数据二维平面的散点图

结论与展望 由于高维数据表达和处理的复杂性妨碍了它的实际应用，为此，我们利用 C++ 的模板技术封装了高维数据复杂的数据结构和算法，并构建了一种新的高维数据类模板及其程序设计框架，用户使用该类模板能方便地为各种特定的高维对象建立数据模型，并通过该高维数据对象的接口实现对高维数据的操作和处理。

在今后工作中，我们考虑将此高维数据类模板移植到 Web Services 环境中，将高维数据"降维"算法封装成 Web Services 组件，通过在 UDDI 发布该组件，使任何 Internet 用户都可以通过 UDDI 搜索到该服务的信息和接口描述，然后用 XML/SOAP 协议发出服务请求，调用和集成相应的"降维"算法，实现对高维数据的处理。

参 考 文 献

1 杨质敏. 高维数据的降维方法研究及其应用. 长沙大学学报，2003,17(2):58～61
2 刘洪波，王秀坤，赵晶. 高维数据空间金字塔技术研究. 计算机工程与应用，2003,39(16):56～57
3 颜雪松，蔡之华. 一种快速聚类高维数据的算法研究. 计算机工程，2003,29(1):131～132
4 肖化昆. 生物制药过程的计算机数字仿真方法与应用. 广东生物数学第四次年会. 广州，2000
5 陈念贻. 模式识别在化工和冶金生产调优中的应用. 广州应用数学成果报告会. 广州，1987