最小二乘法分段直线拟合

田 垅 刘宗田

(上海大学计算机工程与科学学院 上海 200072)

摘 要 曲线拟合是图像分析中非常重要的描述符号。最常用的曲线拟合方法是最小二乘法,然而一般的最小二乘法有一定的局限性,已经有不少学者对其进行了一些改进。进一步对最小二乘法进行改进,提出一种新的分段直线拟合算法来代替多项式曲线拟合,以达到简化数学模型的建立和减少计算的目的,使其能够更好地对点序列进行拟合。

关键词 直线拟合,最小二乘法,分段

中图法分类号 TP301 **文献标识码** A

Least-squares Method Piecewise Linear Fitting

TIAN Long LIU Zong-tian

(School of Computer Engineering & Science, Shanghai University, Shanghai 200072, China)

Abstract Curve fitting is a very important descriptor in image analysis, the most commonly used curve fitting method is least-squares method. But ordinary least-squares method has some limitations, and there are many scholars have made study of improving it. The authors made further improvement on least-squares method and proposed a new piecewise linear fitting algorithm instead of polynomial curve fitting. The new algorithm achieves the goal of simplifying the mathematical model, reducing the calculation, and makes it better to fit point sequence.

Keywords Linear fitting, Least-squares method, Piecewise

1 引言

在工程技术和科学实验中,常常得到两个有函数关系观测量的一系列有序对。如何根据这些有序对来确定它们的函数曲线,这就是实验数据处理中的曲线拟合问题。曲线拟合的应用十分广泛,如在计量经济学领域中可以利用历年的经济统计数据来预测下一阶段的经济发展趋势[1],在医学统计学领域中可以通过发现某种毒物剂量与动物死亡率之间的定量关系来指导医学工作,在传感技术领域可以用于多传感器测量标定[2]。常用的曲线拟合方法是最小二乘法。最小二乘法(又称最小平方法)是一种数学优化技术,它通过最小化误差的平方和寻找数据的最佳函数匹配。利用最小二乘法可以简便地求得未知的数据,并使得这些求得的数据与实际数据之间误差的平方和最小。在利用最小二乘法实现曲线拟合时,虽然能够很好地拟合有序对,并且有很好的数学理论支持[4]和很高的精确度,但是需要进行大量复杂的计算,而且得到的拟合函数往往是高次的,这进一步增加了复杂性。为了简化建立数学模型的算法,减少计算量,本文提出了一种新的利用最小二乘法实现分段直线拟合的方法,以对观测到的有序对进行建模。

2 最小二乘法研究现状

最小二乘法的基本思想是:给定一组实验数据,这些数据往往是有序数对,根据误差平方和最小化原则,找出这些数据的最佳函数匹配。

最小二乘法的数学原理为:给定一组数据$(x_i, y_i)(i=1, 2, \cdots, n)$,设其经验方程为$F(x)$,方程中含有一些待定系数$a_n$。将$(x_i, y_i)$代入方程求差$y_i - F(x_i)$,为了考虑整体的误差,可以取平方和,之所以要平方是考虑到误差正负直接相加可以相互抵消,所以记误差为:

$$e = \sum (y_i - F(x_i))^{\wedge} 2$$

通过求e的极小值可以求出a_n,从而求出该组数据的最佳拟合函数,该函数使得误差平方和最小。

最小二乘法不仅可以用于曲线拟合,也可以用于直线拟合。如果经验方程$F(x)$是线性的,形如$y = a * x + b$,那么利用最小二乘法得到的就是线性回归。

张东林[5]对最小二乘法进行了改进,即将n个有序对$(x_i, y_i)(i=1, 2, \cdots, n)$分成$k$组$N_1, N_2, \cdots, N_k$:

$$(x_{11}, y_{11}), (x_{12}, y_{12}), \cdots, (x_{1N1}, y_{1N1})$$
$$(x_{21}, y_{21}), (x_{22}, y_{22}), \cdots, (x_{2N2}, y_{2N2})$$
$$\cdots$$
$$(x_{k1}, y_{k1}), (x_{k2}, y_{k2}), \cdots, (x_{kNk}, y_{kNk})$$

然后对每组有序对进行线性拟合,得到k条拟合直线$y = g_i(x) = a_i * x + b(i=1, 2, \cdots, k)$。张东林的这种算法虽然能够实现分段直线拟合,并且进一步可以实现曲线拟合,但是其并没有给出k和$N_i(i=1, 2, \cdots, k)$的取值规定。

本文受国家自然科学基金(60975033),国际竹藤组织基础科学研究项目(1632009006)资助。

田 垅(1987-),男,硕士生,主要研究方向为自然语言处理和数据挖掘,E-mail:hubu1931@163.com;刘宗田(1946-),男,教授,博士生导师,主要研究方向为智能信息处理、软件工程等。

谢友宝[3]也对最小二乘法进行了改进。设拟合直线初始起点和初始终点为 $S(x_s,y_s)$、$E(x_e,y_e)$，设终点的下一个点为 $N(x_n,y_n)$，计算线段 ES 和 SN 的夹角为：

$$\angle ESN=\arccos(|SE|2+|SN|2-|EN|2)/(2|SE|*|SN|)$$

式中，$|SE|$、$|SN|$、$|EN|$ 分别为线段 SE、SN、EN 的长度。

观察这个角度是否小于给定的阈值，若小于阈值，则将 E 和 N 往后移一位重新计算，直到大于阈值；否则用最小二乘法计算拟合直线。谢友宝的这种算法虽然实现了有序对的分段直线拟合，并且得到了一种最优的解决方案，但是他没有从整体上来考虑，得到的只是局部最优，而局部最优不一定是整体最优。在此作者进一步对最小二乘进行改进，使其能够从整体上给出最优的分段直线拟合。

3 改进的最小二乘法分段直线拟合

3.1 基本思想

设函数 $f(1,s,e)$ 的返回值为空，其功能是利用最小二乘法对一组有序对 $(x_s,y_s),(x_{s+1},y_{s+1}),\cdots,(x_e,y_e)$ 进行一条直线拟合，得到拟合直线 $y=a_{11}*x+b_{11}$；而函数 $e(1,s,e)$ 是相应的误差平方和，其返回值为 $\sum(y_i-f(1,s,e))\wedge 2$。

类似地，可以定义函数 $e(2,s,e)$ 为利用最小二乘法对该组有序对进行两条直线拟合时得到的误差平方和，且其返回值为对该组有序对进行两条直线拟合时的最小误差；而 $f(2,s,e)$ 的功能是得到两条最佳的拟合直线 $y=a_{21}*x+b_{21}$ 和 $y=a_{22}*x+b_{22}$。

现在利用 $e(1,s,e)$、$f(1,s,e)$ 来得到 $e(2,s,e)$ 和 $f(2,s,e)$，具体步骤如下：

1)将该组有序对分为两组 $(x_s,y_s),(x_{s+1},y_{s+1}),\cdots,(x_i,y_i)$ 和 $(x_{i+1},y_{i+1}),(x_{i+2},y_{i+2}),\cdots,(x_e,y_e)$，其中 $i=s+1,s+2,\cdots,e-2$；

2)分别求出 $e(1,s,i)$ 和 $e(1,i+1,e)$，计算 $t(i)=e(1,s,i)+e(1,i+1,e)$；

3)求出 $t(i)$ 的最小值 $t(\min)$，那么 $e(2,s,e)=t(\min)$；

4)计算 $f(2,s,e)$，具体过程为：

i. 调用函数 $f(1,s,\min)$ 得到第一条最佳拟合直线 $y=a_{21}*x+b_{21}$；

ii. 调用函数 $f(1,\min+1,e)$ 得到第二最佳条拟合直线 $y=a_{22}*x+b_{22}$。

类似地，可以在 $e(k-1,s,e)$ 和 $f(k-1,s,e)$ 已知的情况下，利用递归求出 k 条最佳拟合直线及其相应的最小误差平方和，步骤如下：

1)将该组有序对分为两组 $(x_s,y_s),(x_{s+1},y_{s+1}),\cdots,(x_i,y_i)$ 和 $(x_{i+1},y_{i+1}),(x_{i+2},y_{i+2}),\cdots,(x_e,y_e)$，其中 $i=s+1,s+2,\cdots,e-2$；

2)分别求出 $e(1,s,i)$ 和 $e(k-1,i+1,e)$，计算 $t(i)=e(1,s,i)+e(k-1,i+1,e)$；

3)求出 $t(i)$ 的最小值 $t(\min)$，那么 $e(k,s,e)=t(\min)$；

4)计算 $f(k,s,e)$，具体过程为：

i. 调用函数 $f(1,s,\min)$ 得到第一条最佳拟合直线 $y=a_{k1}*x+b_{k1}$；

ii. 调用函数 $f(k-1,\min+1,e)$ 得到 $k-1$ 条最佳拟合直线 $y=a_{kj}*x+b_{kj}$，其中 $j=2,3,\cdots,k$。

但是如何确定拟合直线的条数呢？一种方法是根据具体的工程需求，人工进行设定；另一种方法是给定一个阈值 β，若连续两次误差平方和的降幅低于这个阈值，则停止拟合的计算，比如，若 $(e(k-3,s,e)-e(k-2,s,e))/e(k-3,s,e)>\beta$ 且 $(e(k-2,s,e)-e(k-1,s,e))/e(k-2,s,e)<\beta$，$(e(k-1,s,e)-e(k,s,e))/e(k-1,s,e)<\beta$，则设定拟合直线的条数为 k。

3.2 实现该算法的部分 java 代码

给定一组有序对 $(x[i],y[i])(i=0,1,\cdots,n-1)$，假设 x 为整型数组，y 为双精度浮点型数组；给定的阈值为 β。

计算误差的 Java 代码：

```
double e(int k,int start,int end){
    double emin=0;
    double fe[]=new double[n];
    //一条直线拟合
    if(k==1){
    double A=0,B=0,C=0,D=0,delta;
    double e1=0;
    double a,b;
    for(int i=start;i<=end;i++)
    {
      A+=x[i]*x[i];
      B+=x[i];
      C+=x[i]*y[i];
      D+=y[i];
    }
  int t=end-start+1;
    delta=A*t-B*B;
    a=(C*t-B*D)/delta;
    b=(A*D-C*B)/delta;
    for(int q=start;q<=end;q++){
    e1+=(y[q]-a*x[q]-b)*(y[q]-a*x[q]-b);
      }
    A=0;B=0;C=0;D=0;
    emin=e1;}else
    {
    for(int i=start+1;i <=end-2*k+2;i++)
      fe[i]=e(1,start,i)+e(k-1,i+1,end);
  int min=start+1;
  double temp=fe[min];
  for(int r=start+1;r <=end-2*k+2;r++) {
      if (temp > fe[r]) {
      temp=fe[r];
      min=r;//间断点
      }
```

```
        emin=temp;
        }
        }
        return emin;
        }
```

计算拟合直线条数的 Java 代码：

```
int k(){
int i=3;
    for(;i<=(2/n);i++){
if ((e(i-2,0,n-1)-e(i-1,0,n-1))/e(i-2,0,n-1)<β && (e
  (i-1,0,n-1)-e(i,0,n-1))/e(i-1,0,n-1)<β){
break;
    }
}
    return i;
}
```

3.3 实例说明

表 1 是某工程中的观测数据。假设阈值为 40%，现利用改进的最小二乘法对其进行分段直线拟合。

表 1

x[ ]	y[ ]	x[ ]	y[ ]
1990	31636.299763	2000	39175.214747
1991	31090.660017	2001	38959.082461
1992	31729.413758	2002	39148.699247
1993	32226.411936	2003	39788.205899
1994	33202.929201	2004	40907.754191
1995	33689.042652	2005	42534.817906
1996	34596.144303	2006	43258.351478
1997	35804.442882	2007	43690.912400
1998	36940.339418	2008	43324.938882
1999	38189.934123	2009	41099.101532

分段直线拟合图如图 1 所示。

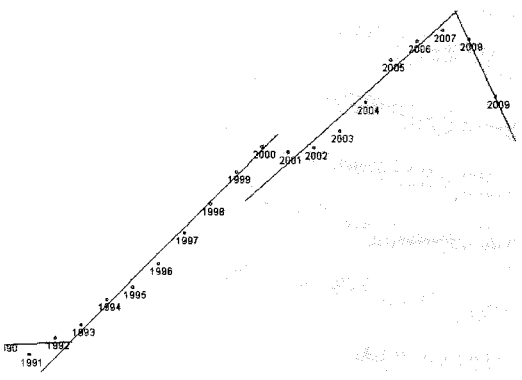

图 1

第一条直线从 $x=1990$ 到 $x=1992$

解析式为：

$y=46.55700000127157*x-61209.52917480469$

误差为：233797.98591108853

第二条直线从 $x=1993$ 到 $x=2000$

解析式为：

$y=1006.4146804752804*x-1973828.8521612259$

误差为：455233.63707434206

第三条直线从 $x=2001$ 到 $x=2007$

解析式为：

$y=898.6216532137929*x-1759653.818239796$

误差为：918928.4677544314

第四条直线从 $x=2008$ 到 $x=2009$

解析式为：

$y=-2225.837350010872*x+4512806.337768555$

误差为：8.902730484240673E-9

误差之和为：1607960.090739871

结束语 以上实例说明，采用改进的最小二乘法进行分段直线拟合，在对实验数据进行处理的过程中，能明显地简化数学模型的建立和减少计算，能够更好地对点序列进行拟合，从而证明本方法是可行的。另一方面，本方法可以自动确定分线段数，自动寻找折点位置，得到整体最优拟合，从而证明本方法是高效的。

但是本方法也有一定的局限性，即对随机变化的数据进行拟合时往往得不到理想的结果，但是对变化比较缓慢的数据可以得到较好的效果，尤其是单调或者近似单调的数据。在实际工作中，试验取得的数据往往是非随机变化的，因此本方法的应用范围还是比较广泛的。

参 考 文 献

[1] 赫尔穆特·鲁克波尔，马库斯·克莱茨希. 应用时间序列计量经济学[M]. 行健，邓可斌，译. 北京：机械工业出版社，2008

[2] 都强，杭柏林. 最小二乘法在多传感器测量标定中的应用[J]. 传感技术学校，2005，18(2)：244-246

[3] 谢友宝. 最小二乘法分段直线拟合[J]. 南昌航空工业学院学报，1992(2)：19-25

[4] 程玉民. 移动最小二乘法研究进展与述评[J]. 计算机辅助工程，2009，18(2)：5-11

[5] 张东林. 分段最小二乘法曲线拟合[J]. 沈阳大学学报，1994(2)：80-83

[6] Plackett R L. The discovery of the method of least squares[J]. Biometrika，1972，59：239-251

[7] Bates D M，Watts D G. Nonlinear regression analysis and its applications[M]. New York：Wiley，1988

[8] Harper H L. The method of least squares and some alternatives [M]. Part I，II，II，IV，V，VI. International Statistical Review，42，147-174；42，235-264；43，1-44；43，125-190；43，269-272；44，113-159

[9] Shimizu T，Kyochi S，Ikehara M. A Design of Dual-Tree Complex Wavelet Transform Based on Least Squares Method[C]// Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop. 2009：576-581