

改进的数据流缺失数据处理算法

张宙锋^{1,2}, 张 磊¹, 王志超¹

(1 中国矿业大学 计算机学院, 江苏 徐州 221116; 2 潞安集团技术中心, 山西 长治 046204)

摘 要: 数据流预处理主要是在原始观测数据的基础上进行, 包括对原始监测到的数据集中的缺失数据进行插补或删除, 是数据流预测过程中一个重要环节, 是数据流应用中必不可少的组成部分. 数据流预处理技术可以改进监测数据流的质量, 从而有助于提高其后的处理过程的精度和性能.

关键词: 数据流; 检测; 缺失数据

中图分类号: TP312

文献标识码: A

文章编号: 1000-7180(2012)03-0055-05

Improved Missing Stream Data Interpolation Algorithm

ZHANG Zhou-feng^{1,2}, ZHANG Lei¹, WANG Zhi-chao¹

(1 School of Computer Science, China University of Mining and Technology, Xuzhou 221116, China;

2 Centre of Technology, Lu'an Group Co. Ltd., Changzhi 046204, China)

Abstract: The preprocessing of data stream is mainly carried out on the raw data, including interpolating the missing data and excluding the useless data on the monitoring original data set. It is an important and essential part on the data stream preprocessing. Data stream preprocessing can improve the quality of monitoring data stream, and can improve the accuracy and performance of the subsequent processing.

Key words: data stream; monitoring; missing data

1 引言

在保证数据流质量的前提下, 结合 Pipeline 数据处理方法, 文献[1]提出了一种数据流上的数据处理算法, 通过计算信任度和覆盖率来评估流的质量, 主要解决数据流环境下的数据漏读和数据多读问题. Jeffrey SR 等^[2]提出一种 ESP 机制, 可以清洗来自不同接收器的数据, 并且可以针对各类型缺失数据的特点进行清洗. Khousainova N 等^[3]根据具体应用, 提出了一种主要基于概率性模型的数据处理算法, 主要解决数据漏读和数据错读问题. 文献[4]提出了利用三次参数样条函数对缺失数据进行修补的方法. Gonzalez H, 等^[5]提出了一种通过统计学习和工作流建模来进行数据挖掘的方法, 相关的模型可以用于简单情况的以漏读数据为核心的处理. 但没有考虑数据流应用环境下的数据漏读率等重要的因素, 并且它是基于数据库, 不是针对数据流进行处

理的.

2 带有空值检验的数据流缺失数据处理模型

2.1 处理模型

针对连续缺失和随机不连续缺失这两种数据流环境下的数据缺失情况, 本文提出了一种新的带有缺失项检验的数据流缺失数据处理模型, 如图 1 所示. 针对随机不连续缺失的情况, 由于前后数据的时间间隔较短, 采用计算前后数据平均值作为缺失数据项的插补值. 针对数据连续缺失的情况, 由于缺失数据项前后的时间跨度比较大, 本文提出了一种基于 EM 的缺失数据插补算法. 算法有两个特点: (1) 设定 EM 算法的最高迭代次数, 当 EM 算法的迭代次数超过初始给定的阈值时, 即停止迭代, 用最后一次计算出的结果作为缺失数据项的插补值; (2) 重新设置 EM 算法的计算缺失数据项的触发条件, 对于连续数据缺失的情形, 给定一个阈值 M , 当连续

缺失的数据项数目小于 M 时, 仅仅利用平均值插补方法计算缺失数据项, 只有当连续缺失的数据项数目大于 M 时, 才利用 EM 算法计算缺失数据项的替代值. 插补效果比较如图 2 所示.

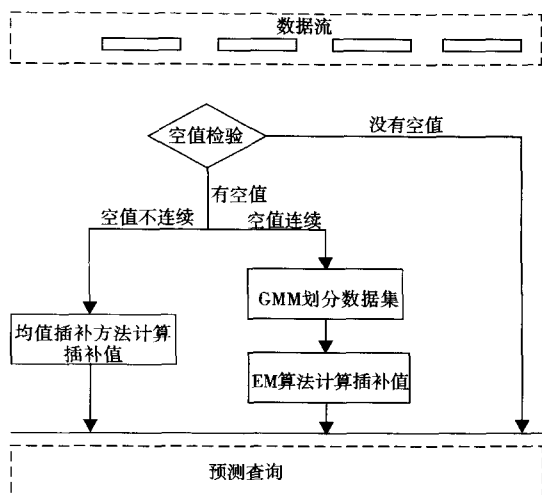


图 1 带有空值检验的数据流处理模型

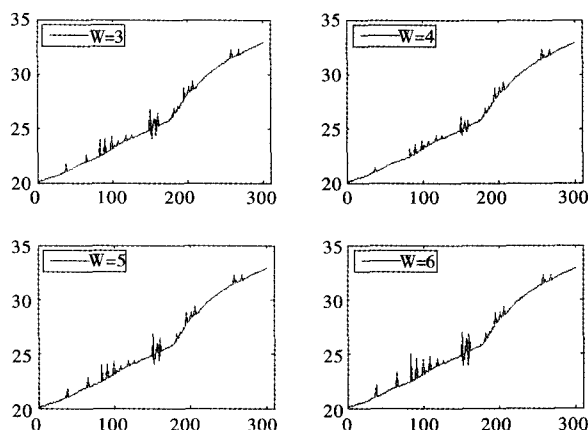


图 2 温度上的插补效果比较

2.2 数据流环境下的缺失数据插补算法

2.2.1 算法描述

本文数据处理算法基于固定窗口宽度的滑动窗口模型, 给定窗口宽度为 N . 令数据缺失数据阈值为 W , 数据流正常到达数据窗口的时间间隔 t_0 , 计算前后到达数据窗口的两条数据 (S_{n+1}, t_{n+1}) 与 (S_n, t_n) 之间的时间间隔 $T = t_{n+1} - t_n$, 比较 T 与 $W * t_0$ 的大小, 共分三种情形: (1) 当 $T < 2 * t_0$ 时, 数据流监测正常, 数据集中无缺失项; (2) 当 $2 * t_0 \leq T \leq W * t_0$ 时, 即 (S_{n+1}, t_{n+1}) 与 (S_n, t_n) 之间存在缺失数据项, 此时采用平均值方法计算缺失数据项, 即 (S_{n+1}, t_{n+1}) 与 (S_n, t_n) 之间的缺失项均用 $(S_{n+1} + S_n) / 2$ 代替; (3) 当 $T > W * t_0$ 时, 即缺失数据项的数目超出了阈值范围, 采用基于 EM 的插补算法计算插补

值, 算法具体分为两个步骤: GMM 划分初始数据集和 EM 方法计算插补值. 为了描述方便, 定义当前窗口中的数据集为 X , 由于窗口宽度为 N , 则整个数据集的数目为 N , 其中的 m 组连续缺失的数据项定义为 $x_i (i=1, 2, \dots, m)$.

GMM 划分数据集:

(1) 扫描数据集 X , 分别以数据集中的 m 组连续缺失的数据集为中心, 对原始数据集进行初始划分, 数据集 X 被划分成 m 组正态分布的数据组, 分别记为 $X_1, X_2, \dots, X_m$, 则 $X_1, X_2, \dots, X_m$ 符合混合高斯分布.

(2) 对于数据集 $X_1, X_2, \dots, X_m$, 分别计算各组数据的均值向量 $e_j (j=1, 2, \dots, m)$, 协方差矩阵 $\Sigma_j (j=1, 2, \dots, m)$, 根据各组数据的数目 N_j 所占整个数据集 N 的比重, 计算出各组数据在整个数据集上的初始权重 $\alpha_j^0 = N_j / N (j=1, 2, \dots, m)$, 由于监测数据是一维的, 故 $\Sigma_j = 1 (j=1, 2, \dots, m)$, 因此在以下的计算中 Σ_j 均可省略.

EM 方法计算插补值: 给定初始迭代次数 $F=0$, 精度阈值 ϵ , 迭代次数阈值 F_0 .

(1) E一步: (期望步) 计算隐含变量数据的期望值. 用随机变量 C 指示数据成分, 则概率 $P_{ij} = P(C=i | x_j)$ 表示数据 x_j 由成分 i 产生的概率, 也即由第 j 高斯分布产生的概率. 由贝叶斯公式, 有 $P_{ij} = P(x_j | C=i) P(C=i)$, 其中 $P(x_j | C=i)$ 即 x_j 在第 i 组高斯分布中的概率. 而 $P(C=i)$ 是第 i 高斯分布的权重参数. 应用到本例定义的 GMM 中, 表示如下:

$$P(j | x) = \frac{\alpha_j G(x, e_j, \Sigma_j)}{P(x, \Theta)}, \quad j = 1, 2, \dots, M$$

(2) M一步: (极大化步) 主要通过求解对数似然方程, 计算出期望值到达极大值点时新的均值 e_j , 权重 α_j , 用于下次叠代, 其中 $\sum_j = 1$.

$$e_j = \frac{\sum_{i=1}^M P(j | x_i) x_i}{\sum_{i=1}^M P(j | x_i)} = \frac{1}{\alpha_j * M} \sum_{i=1}^M P(j | x_i) x_i$$

$$\alpha_j^{(t+1)} = \frac{1}{M} \sum_{i=1}^M \frac{\alpha_j^{(t)} G(x_i, e_j, \Sigma_j)}{\sum_{j=1}^M \alpha_j^{(t)} G(x_i, e_j, \Sigma_j)}$$

$$= \frac{1}{M} \sum_{i=1}^M P(j | x_i)$$

(3) 首先判断迭代次数 F 是否等于 F_0 , 是则停止迭代, 以最近一次计算得到的 $e_j (j=1, 2, \dots, m)$

作为缺失数据项的插补值,否则按照预先设定的阈值条件 ϵ ,测试预测结果是否满足初始设置的循环迭代条件 $\|\alpha_j^{(i+1)} - \alpha_j^{(i)}\| \leq \epsilon$,如果不满足条件,则转第(2)步,迭代次数 F 加 1,如果满足条件,则迭代停止,输出最终参数 α_j 和 $e_j (j=1,2,\cdots,m)$,此时 $e_j (j=1,2,\cdots,m)$ 即可作为数据集中缺失点的插补值.

2.2.2 算法复杂度分析

由于固定窗口的宽度为 N ,假设当前窗口的数据缺失比率为 $P_0 (0 < P_0 < 1)$,则窗口中缺失的数据项数目为 $N \times P_0$. 假设当前窗口上所有的缺失数据项均为非连续缺失的,适用于平均值插补算法,此时算法的时间复杂度为 $O(N \times P_0)$. 当窗口上存在连续缺失的情形时,根据预先设定的缺失阈值 W ,当连续缺失的数据小于 W 时,此时采用均值插补的方法对缺失数据进行平滑处理,当连续缺失的数据数目大于 W 时,利用 EM 方法首先将数据集划分为不多于 NP_0/W 个正态分布数据子集,各子集上的数据不超过 $N(NP_0/W) = W/P_0$ 条数据,分别对各子集进行 EM 迭代,根据初始设定的迭代次数阈值 F_0 ,则计算出数据集上全部缺失数据项的插补值的时间复杂度为 $O\left(2 \times \frac{NP_0}{W} \times F_0 \times \frac{W}{P_0}\right) \rightarrow O(2NF_0)$.

由此,可以得出该算法的时间复杂度为 $O(2F_0N)$,与缺失阈值 W 无关. 数据流环境下,根据单个数据到达窗口的时间间隔 t_0 ,则新数据充满整个窗口的时间为 $N \times t_0$. 因此,通过调整算法的迭代次数阈值 F_0 ,可以控制算法的运算时间,从而在小

于一个时间窗口的范围内完成缺失数据的插补.

3 实验及结果分析

3.1 实验说明

使用 Java, NetBeans6.1 环境下,在 Windows XP,CPU2.79 GHZ,内存 2G 的计算机上完成实验. 如表 1 所示. 实验数据是采用美国 NIST 的火灾模拟软件 FDS4.0 进行火灾模拟得到的数据集. 传感器采样频率 $f=0.2$ s,共采集了 30 分钟内的 10000 条火灾发生现场监测数据. 表中的缺失数据是利用随机数方法,将一部分数据从原始数据集中剔除掉.

3.2 不同迭代阈值条件下的插补效率比较

本文将窗口的宽度设定为 300,此时窗口的时间跨度为 60 秒,设定 EM 方法初始循环迭代条件 $\epsilon=0.01$,为验证算法在不同迭代阈值条件下的时间效率,算法缺失阈值 $W=5$,在此基础上分别将 EM 算法的迭代次数阈值设置为 5,6,7,8. 为验证插补的效果,文中给出了衡量和比较插补误差的计算方法. 令 Y 表示完全数据资料集; Y_{mi} 为 Y 中人为删除的缺失项; Y_{α} ,为 Y 中与 Y_{mi} 对应的原始观测值; $Y_{\hat{\mu}}$,为经过计算后的缺失替代数值. 给出插补效果误差的计算公式:

$$E = \sum_{i=1}^n (Y_{\hat{\mu}} - Y_{\alpha})^2$$

应用上述公式,计算四次迭代阈值条件下的误差效果如表 2 所示.

表 1 FDS4.0 火灾模拟现场数据流

TIME /s	TEMP /C	RAD LOSS /kW	CONV LOSS /kW	COND LOSS /kW	BURN RATE /kg/s
3.00259	20.1329	219.053	-18.5067	268.971	0.050812
3.27947	20.1533	228.822	-8.24093	245.317	NULL
3.52559	20.1741	248.671	10.0905	254.254	0.051005
3.75632	NULL	256.229	16.0257	271.685	0.051045
4.02704	20.2094	252.114	NULL	NULL	0.051072
4.26393	20.2217	252.238	NULL	280.826	0.051089
4.50082	20.239	268.004	NULL	285.372	0.051099
...	...	...	...	...	...
891.207	48.6285	13.8383	-23.1973	23.8686	0.59285
891.444	48.6301	13.82	-23.1921	23.8683	0.61258
891.681	48.6316	13.7953	-23.1845	23.8614	0.61328
891.918	NULL	13.8411	-23.1983	23.8658	NULL
892.156	48.6348	NULL	-23.1935	NULL	0.61585
892.393	NULL	13.7856	-23.2046	23.8608	0.61658

表 2 不同迭代阈值条件下的计算效果比较

阈值(F_0)	计算时间(s)	误差值
5	8.7	55.541
6	10.43	43.57
7	12.19	40.574
8	11.72	39.682

表 2 给出了四次实验的结果. 从表中可以看出, 随着迭代次数阈值的增加, 算法所耗时间也相应增长. 迭代阈值为 8 时, 算法所耗时间减少, 这是因为当前窗口中的大部分连续缺失项小于 8, 根据算法的处理策略, 采用均值方法对大部分的缺失数据进行插补, 因而算法的时间效率有所提升. 实验表明, 算法可以在小于一个时间窗口(60 s)的范围内完成缺失数据的计算, 达到了预期的目标.

对于不同阈值条件下的误差效果, 从表中可以看出随着迭代次数的增长, 算法的误差精度相应的提高, 得出本算法的最优迭代阈值为 7.

3.3 不同缺失阈值条件下的插补效果比较

为了验证在不同连续缺失数目阈值条件下算法的插补效果, 根据上文计算结果, 首先将迭代阈值 F_0 设定为 7, 在此基础上, 分别将连续缺失数目阈

值 W 设置为 3, 4, 5, 6 在整个数据集上做了四次插补实验. 如表 3 所示.

表 3 不同缺失阈值条件下的插补效果比较

阈值(W)	误差值
3	65.541
4	39.570
5	75.574
6	82.621

从表 3 中, 可以得出在三次实验中当缺失阈值为 4 时, 算法的插补效果最好. 此时算法得出的插补值精度最高, 插补后的数据曲线与原始数据曲线拟合效果最好. 这是因为当给定的缺失阈值过小时, 原本可以直接应用平均值插补的缺失项在算法的判断策略下被应用 EM 方法计算的全局最优解替代, 此时反而降低了插补精度, 同时提高了算法的时间复杂度. 在本次实验中, 当缺失阈值为 4 时, 算法的性能达到最优.

将缺失阈值 W 设置为 4, EM 算法的迭代阈值 F_0 设置为 7, 收敛阈值 $\epsilon=0.01$, 窗口宽度设定为 300, 在整个数据集上做了一次完整的插补实验. 表 4 给出了经过插补后的数据集.

表 4 经处理后的 FDS4.0 火灾模拟现场数据流

TIME /s	TEMP /C	RAD LOSS /kW	CONV LOSS /kW	COND LOSS /kW	BURN RATE /kg/s
3.00259	20.403	219.053	-18.5067	268.971	0.050812
3.27947	20.426	228.822	-8.24093	245.317	0.050933
3.52559	20.45	248.671	10.0905	254.254	0.051005
3.75632	20.477	256.229	16.0257	271.685	0.051045
4.02704	20.498	252.114	10.0919	280.015	0.051072
4.26393	20.514	252.238	1.80229	280.826	0.051089
4.50082	20.535	268.004	3.5516	285.372	0.051099
...	...	...	...	...	...
891.207	48.6285	13.8383	-23.1973	23.8686	0.59285
891.444	48.6301	13.82	-23.1921	23.8683	0.61258
891.681	48.6316	13.7953	-23.1845	23.8614	0.61328
891.918	48.6332	13.8411	-23.1983	23.8658	0.61467
892.156	48.6348	13.8131	-23.1935	23.8601	0.61585
892.393	48.6363	13.7856	-23.2046	23.8608	0.61658

4 结束语

平均值插补是一种非常简单有效的缺失数据插

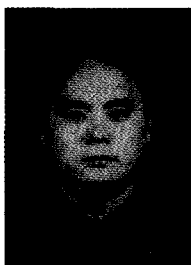
补方法, 在数据流环境中, 单个缺失数据项利用平均值方法可以有效的完成数据集的平滑处理. 空值前后的时间跨度较大, 使用 EM 算法可以有效完成.

用样本总体最优解来完成缺失数据的插补,更加贴近样本总体的实际分布. 为降低 EM 算法的时间复杂度,本文提出一种新的数据流缺失数据处理算法. 该算法根据数据流数据缺失的情况,设置合理的阈值条件,当连续缺失的数据项超出阈值条件时,才采用 EM 算法计算缺失数据项的插补值,否则直接利用平均值方法完成数据集的处理. 该算法结合了平均值插补和 EM 算法计算插补值的优点,可以有效的满足数据流环境下大数据量和高速处理的需要,同时进一步提高了插补效果,提高了插补精度. 最后,通过实验证明了该算法有效.

参考文献:

- [1] Jeffrey S R, Alonso G, Franklin MJ, et al. A pipeline framework for on line cleaning of sensor data streams[C]// Int'l Conf. on Data Engineering. Atlanta, 2006:140-142.
- [2] Jeffery S R, Garofalakis M, Franklin M J. Adaptive cleaning for RFID data streams[C]// VLDB 2006. Korea, 2006: 163-174.
- [3] Zhuang Y Z, Chen L. In-Network outlier cleaning for data collection in sensor networks[C]// VLDB 2006. Korea, 2006: 41-48.
- [4] 张春生,赵凌琪,庄丽艳. 部分缺失数据的三次参数样条函数修补方法 [J]. 微电子学与计算机, 2010, 27(5):193-196.
- [5] Gonzalez H, Han JW, Li XL. Mining compressed commodity workflows from massive RFID data sets [C]// JIKM 2006. USA: Arlington, 162-171.

作者简介:



张宙锋 男,(1975—),硕士研究生,工程师. 研究方向为数据挖掘、数据流.

(上接第 54 页)



廖祖华 (1957—), 硕士生导师,教授. 研究方向为人工智能、模糊代数.

曹 姝 女,(1987—),硕士研究生. 研究方向为人工智能、模糊代数.

王德江 (1989—),硕士研究生. 研究方向为人工智能、模糊代数、图像处理.

刘春芝 女,(1988—),硕士研究生. 研究方向为人工智能、模糊代数.