文章编号：2095-2163(2019)04-0001-06 中图分类号：TP391.41 文献标志码：A

基于能量过滤的不确定时间序列数据清洗方法

孙纪舟，李建中
（哈尔滨工业大学 计算机科学与技术学院，哈尔滨 150001）

摘 要：精确度是数据科学领域研究的重要方面，对后续数据处理等过程都有至关重要的影响。利用多个传感器返回的多个时间序列可提升时间序列数据的精确度，称为不确定时间序列，这多个时间序列样本在真实数据上下随机波动。已有关于时间序列的研究大多直接在不确定时间序列上提出新算法，其缺点是算法复杂度通常较高，直接对不确定时间序列进行清洗，获得尽可能接近真实的数据有重要意义。本文提出基于能量过滤的方法对不确定时间序列进行清洗，实验结果表明与已有方法相比，本文方法在效果和效率上都更优。

关键词：不确定时间序列；能量过滤；数据清洗

Uncertain time series data cleaning based on energy filter

SUN Jizhou, LI Jianzhong
(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

【Abstract】 Large scale time series data are generated and processed in many applications, such as data collected by sensors. Due to environmental interferences and low precision of the sensors, the collected data are not usually accurate. Accuracy is an important aspect of data science area, which plays a key role in the subsequent data processing tasks. To improve data quality, multiple sensors are often deployed to collect data at the same location. The time series samples returned by the sensors are called uncertain time series data. Existing research often designs new algorithm for old problems on uncertain time series, the drawbacks are low efficiencies. Cleaning the uncertain data to get a single series close to the truth as much as possible is of great significance. Traditional algorithms designed for certain time series can be applied directly. A power filter based method to clean uncertain time series is proposed in this paper, experimental results show that the proposed method is better in both effectiveness and efficiency.

【Key words】 uncertain time series; power filter; data cleaning

0 引 言

时间序列数据在日常生活和工业生产中无处不在，例如气象学中的温度、湿度、风速、PM2.5；医学中的心跳、血压、体温；以及经济学中的股票指数、恩格尔系数以及其它描述宏观经济形势的指数等。这些数据都是随时间变化的数值型数据。由于环境干扰、传感器的精度不够、获取数据时的舍入等原因，时间序列数据通常是不精确的，距离真实数据总有一些误差。而这些误差往往给人们的日常生活、医疗中的病情诊断及监控以及政府部门的决策等带来负面影响。

为了尽可能降低误差带来的影响，常用的解决方法就是对同一时间序列数据采集多个样本，每个样本都在真实数据周围随机的上下波动，对这些样本求平均值，或者直接在这些样本上设计新算法，都能在一定程度上解决误差带来的影响。求平均值的方法最简单快速，但结果精确度不够高；设计新算法的思路能够获得更高的精度，但往往有着很高的时间复杂度。

结合时间序列平滑的特性以及随机噪声的波动特性，本文给出一种基于能量过滤的时间序列清洗算法。根据给定的时间序列样本，计算出数据中噪声所占能量的比重，根据这个比重找出一个频率阈值，并将傅里叶变换之后高于该阈值的部分过滤掉，所得结果更加平滑且接近真实数据，在 Top-k 查询问题上和已有算法做了实验对比，结果显示在效果上本文算法较好，而时间效率上本文算法远远优于已有算法。

1 问题描述

1.1 时间序列

时间序列 S 是一个长度为 n 的有序数据序列 $< s_1, s_2, \cdots, s_n >$，其中 $s_i (1 \leqslant i \leqslant n)$ 是数值型数

基金项目：国家自然科学基金(61190115,61033015)；国家重点基础研究发展计划(973)(2012CB316200)。

作者简介：孙纪舟(1985-)，男，博士研究生，主要研究方向：数据质量、海量计算；李建中(1950-)，男，教授，博士生导师，主要研究方向：并行数据库、传感器网络、海量数据管理等。

收稿日期：2018-10-11

据,表示第 i 个时刻对应的数值如温度、湿度、股票价格等,也可表示为 $S[i]$。

在时间序列数据的相关应用中,最为广泛的操作是计算时间序列之间的距离。给定 2 个长度为 n 的时间序列 S 和 T, 计算 S 和 T 之间的距离度量: $Dist(S,T)$, 针对应用场景的不同,对 S 和 T 距离的度量方式有很多, 常用的包括动态时间规整(DTW)、欧式距离等。其中欧氏距离是应用最为广泛的时间序列相似度度量方法。其形式化定义如下:

$$Dist(S,T) = \sqrt{\sum_{i=1}^{n} (S[i] - T[i])^2}.$$

其中, $Dist(S,T)$ 越小,表示 S 和 T 的距离越小,其相似度也越高。

1.2 不确定时间序列

在很多实际情况中,收集到的数据往往是不精确的,比如采集温度数据的传感器,本身有一定的误差,为降低误差,对同一时刻的数据收集多个数据样本,以提高测量精度。因此本文给出的不确定时间序列模型描述如下:

不确定时间序列 X 是一个有序集合 $< x_1, x_2, \cdots, x_n >$,其中$x_i(1 \leqslant i £ \leqslant n)$ 是数值型数据集合,表示第 i 个时刻对应数值的所有样本值,$|x_i| = m$ 是样本大小,x_i 也可表示为 $X[i]$。x_i^j 表示第 i 个时刻的第 j 个样本值。 为便于理解,将 i 时刻某一样本的可能取值看成一个连续型随机变量X_i,X_i 的概率分布和当前时刻的真实值s_i 以及数据收集器(如温度传感器等) 的误差有关, 因此可表示为:

$$X_i = s_i + Y_i.$$

其中: Y_i 是一个随机变量,表示第 i 时刻的误差分布。为简化问题,本文给出 2 个合理的假设:

(1)不同时刻值的误差是独立同分布的随机变量;

(2) Y_i 的期望值为 0,即数据收集器自身没有系统偏差。

1.3 不确定时间序列的清洗

关于不确定时间序列的已有研究中,都致力于提出新的模型和算法对不确定时间序列数据进行搜索、聚类和 Top-k 查询等。而相关问题在确定时间序列上的研究已经十分成熟,为了使这些方法能够直接用在不确定时序数据上,本文主要研究如何对不确定数据进行清洗(或者还原),使之变为尽可能接近真实数据的确定时间序列。下面给出不确定时间序列的清洗问题。

对于时间序列 $S = < s_1, s_2, \cdots, s_n >$, 不确定时间序列 $X = < x_1, x_2, \cdots, x_n >$ 是对 S 的观察值,其中 x_i 是对 s_i 进行观测得到的 m 个样本值。在 X 已知, S 未知的条件下,计算 S 的估计值 $\hat{S} = < \hat{s}_1, \hat{s}_2, \cdots, \hat{s}_n >$, 使得 $\hat{S}$ 尽可能的接近 S, 即 $Dist(S, \hat{S})$ 尽可能小。

之所以要最小化估计值和真实值之间的欧氏距离,是因为欧氏距离是评估时间序列相似度中最常用的方法。若要查询序列 QS 到 S 之间的距离 $Dist(QS,S)$,考虑用 $Dist(QS,\hat{S})$ 去近似,由于欧氏距离满足三角不等式,即对于任意时间序列 S_1 , S_2 和 S_3 , $Dist(S_1,S_2) + Dist(S_2,S_3) \geqslant Dist(S_2,S_3)$ 恒成立, 不难得出 $Dist(QS,S) \in [Dist(QS,\hat{S}) - Dist(S,\hat{S}), Dist(QS,\hat{S}) + Dist(S,\hat{S})]$, 很明显, $Dist(S,\hat{S})$ 越小,该区间就越小,所得到的结果也就越精确。

2 基于能量过滤的清洗方法

为了对不确定序列进行估计,最直观的方法是直接对每个时间点的样本求均值 $\overline{x_i}$,并用 $\overline{x_i}$ 作为相应的估计值 $\hat{s}_i$。 然而这种方法单独对每个时间点进行计算,忽略了相邻时间点数据的相关性,实验验证部分也证实该方法并不能够很好地对数据进行还原。

由于数据点之间的相关性在频域表现比较明显,因此本文考虑在频域进行降维,从而达到清洗数据的目的。其直观思想是,时间序列数据在频域上分布极不均匀。即有些频率上的数据分布很集中(高能区域),而有些频率上只有很少数据信息(低能区域),而不确定数据中的噪声在各个频率上的分布相对均匀。因此,在低能区域,噪声数据占据主导地位,直接将其舍弃掉虽然会丢失一部分有用信息,但同时丢掉了更多的垃圾信息,使得整体的数据质量得到提升。该方法的优点主要包括:

(1)大大减少了数据量,每个时间点的数据由 m 维降低到 1 维,并且在频域上只需要保留很少的数据(例如在实验中, 长度为 2^k 的数据在频率域只需要保留 100 个左右的数据点);

(2)大大提升了数据质量,通过自适应的选取一个能量阈值,本文的方法能够去掉尽可能多的噪声,保留尽可能多的有用信息,从而使最终的估计结

果尽可能地接近真实数据，实验部分也对此进行了验证。

2.1　离散傅里叶变换

将时间序列转换到频域，需要用到的是离散傅立叶变换(DFT)。给定时间序列 $S = < s_1, s_2, \cdots, s_n >$，$S$ 的离散傅立叶变换是一个复数序列 $F = < f_1, f_2, \cdots, f_n >$，其中：

$$f_k = 1/\sqrt{n} \sum_{i=1}^{n} s_i \exp(-j2\pi ik/n),$$

其中，j 是虚数单位 $\sqrt{-1}$。反过来，S 可以通过对 F 进行逆变换得到：

$$s_i = 1/\sqrt{n} \sum_{k=1}^{n} f_k \exp(j2\pi ik/n),$$

傅立叶变换有快速算法(FFT)，其时间复杂度为 $O(n\log(n))$。对于复数 $c = a + jb = A\exp(j\varphi)$，$A$ 称之为复数的幅值，φ 称之为相位，其能量 $E(c) = a^2 + b^2$，复数序列的能量 $E(F) = \sum_{k=1}^{n} E(f_k)$。对于离散傅立叶变换，有如下重要定理：

定理 1(Parseval)　如果 F 是 S 的离散傅立叶变换结果，那么：

$$\sum_{i=1}^{n} E(s_i) = \sum_{k=1}^{n} E(f_k).$$

Parseval 定理的直观含义是，序列数据在经过离散傅立叶变换之后，其能量保持不变。另外，离散傅立叶变换是一个线性变换。

定理 2　傅立叶变换具有线性性质：

(1)若 S_1 的傅立叶变换结果是 F_1，S_2 的傅立叶变换结果是 F_2，那么 $S_1 + S_2$ 的傅立叶变换结果是 $F_1 + F_2$；

(2)若 S 的傅立叶变换结果是 F，且 a 为任一实数，那么若 aS 的傅立叶变换结果是 af。其中 $S_1 + S_2$ 表示 2 个序列对应元素相加，aS 表示序列的每个元素乘以 a。

结合上述 2 个定理，容易得出结论：如果 S_1 的傅立叶变换结果是 F_1，S_2 的傅立叶变换结果是 F_2，那么：

$$Dist(S_1, S_2) = E(S_1 - S_2) = E(F_1 - F_2) = Dist(F_1, F_2).$$

该结论说明，2 个时间序列的欧式距离等于他们在频域的欧式距离。

2.2　基于能量过滤清洗时间序列数据

对不确定时间序列 X 进行均值处理后，初步得到一个比较接近真实值的序列 $\bar{X} = < \bar{x}_1, \bar{x}_2, \cdots, \bar{x}_n >$，其中，$\bar{x}_i = 1/m \sum_{x \in x_i} x$，即对每个时刻的样本数据分别求均值。这时 $\bar{X}$ 可看成是真实数据 S 和噪声数据 Err 的叠加：

$$\bar{X} = S + Err,$$

对 $\bar{X}$ 进行离散傅立叶变换得到 $F_{\bar{X}}$，且根据定理 2，有 $F_{\bar{X}} = F_S + F_{Err}$，其中 F_S 和 F_{Err} 分别是 S 和 Err 的离散傅立叶变换结果。对于 $F_{\bar{X}}$ 中的 n 个复数按照幅值绝对值降序排列，得到长度为 $2n$ 的实数序列 $< r_1, r_2, \cdots, r_{2n} >$，其中，每个元素 r_i 都由 2 部分组成：真实值 s_i 和噪声 ε_i。把其都看作分别从随机变量 R，S 和 Er 中得到的样本，则对于数据能量的期望，有：

$$E(R^2) = E((S + Er)^2) = E(S^2) + E(Er^2) + E(SEr),$$

由前面的假设，知道噪声的均值为 0，且和真实数据无关，上式变为：

$$E(R^2) = E(S^2) + E(Er^2) + E(SEr) = E(S^2) + E(Er^2) + E(S)E(Er) = E(S^2) + E(Er^2).$$

即在某个频率上，脏数据的能量的期望等于真实数据能量期望与噪声能量期望之和。

对于某频率的数据，如果将其舍弃掉，将导致与真实数据之间 $E(S^2)$ 的能量差；如果将其保留，则将导致与真实数据之间 $E(Er^2)$ 的能量差。因为能量和欧式距离之间有平方的关系，因此要使这个能量差尽量小。如果 $E(S^2)$ 未知而 $E(Er^2)$ 已知，那么只要某频率上，脏数据的能量大于 2 倍的 $E(Er^2)$，就说明真实数据能量的期望大于噪声能量的期望，将该频率上的值保留，否则说明真实数据能量的期望小于噪声能量的期望，真实数据被噪声淹没，则将该频率上的值舍弃。下面讨论如何估计噪声期望。

2.3　噪声能量的估计

由于不同时刻的数据都是由同一个传感器收集的，因此不同时刻的随机噪声也是独立同分布的。每个时刻有 m 个样本，均由随机变量 $s + Ns$ 中采样得到，其中 s 是真实值但未知，随机变量 Ns 是传感器的随机误差。由于 s 是常数不影响方差，因此 $s + Ns$ 和 Ns 的方差相等，由概率论知识可知，m 个样本的样本方差是对 $s + Ns$ 方差的无偏估计，即是对 Ns 方差的无偏估计。由于时间序列很长，因此在每个时间点上的数据估计 Ns 并求平均，根据大数定律容易得出，如此求得的方差几乎等于传感器随机误差的方差：

$$Var(Ns) \approx 1/n \sum_{i=1}^{n} 1/(m-1) \sum_{j=1}^{m} (x_i^j - \bar{x}_i)^2.$$

同时,由方差和期望的性质可知: $Var(Ns) = E(Ns^2) - E^2(Ns)$, 而传感器没有系统偏差,即 $E(Ns) = 0$, 从而可得 $E(Ns^2) = Var(Ns)$, 其含义是,传感器误差的方差等于其所产生的噪声数据的能量。因为能量过滤是在求均值之后的序列数据上进行的,因此可以得知,每个时间点上 $\bar{X}$ 中噪声能量的期望是:

$$Var(Ns)/m \approx 1/n \sum_{i=1}^{n} 1/(m^2 - m) \sum_{j=1}^{m} (x_i^j - \bar{x}_i)^2. \tag{1}$$

因为噪声数据在频域是均匀分布的,容易得出每个频率上噪声能量的期望也是 $Var(Ns)/m$, 平均到复数的实部和虚部,各占一半。因此 $E(Er^2) = Var(Ns)/2m$, 能量过滤的阈值设为 $th = 2E(Er^2) = Var(Ns)/m$。

2.4 算法

至此,可给出基于能量过滤的时间序列清洗算法:

算法 1 FilterClean 算法。

输入 长度为 n,样本数位 m 的不确定时间序列 X。

输出 长度为 n 的确定时间序列 $\hat{S}$,使其与真实数据的距离尽可能小。

(1) 对 X 每个时间点上的数据分别求平均,得到 $\bar{X}$;

(2) 对 $\bar{X}$ 进行快速傅立叶变换,得到频域序列 F;

(3) 根据式(1) 求出能量过滤的阈值 th;

(4) 对 F 中每个频率上的复数;

(5) 如果复数实部的平方小于 th,则将其实部设置为 0;

(6) 如果复数虚部的平方小于 th,则将其虚部设置为 0;

(7) 得到过滤后的频域序列 F';

(8) 对 F' 进行快速傅立叶逆变换,得到时域序列 $\hat{S}$;

(9) 返回 $\hat{S}$。

算法复杂度分析: 第(1)步时间复杂度为 $O(mn)$, 第(2)步时间复杂度为 $O(n\mathrm{Log}(n))$, 第(3)步时间复杂度为 $O(mn)$, 第(4)~(7)步时间复杂度 $O(n)$, 第(8)步时间复杂度为 $O(n\mathrm{Log}(n))$, 因此,算法的时间复杂度是 $O(n(\mathrm{Log}(n) + m))$。

3 实验验证

最后在真实数据集和合成数据集上对本文算法和其它算法做一对比。

3.1 实验环境

本文算法代码用 JAVA 语言实现,硬件环境是主频 3.60GHz 的 8 核 Intel i7 处理器,内存大小为 8GB,硬盘大小 1TB 的台式机,底层操作系统是 Windows 7。

3.2 实验数据

本实验采用的数据集为 UCR 数据集,UCR 是时间序列数据研究中最常用的数据集,样本及噪声的生成均采用文献[1]中的方法。

3.3 算法对比

本实验主要与一个最近的关于不确定时间序列数据上 Top-k 查询的算法[1] Holistc-PkNN 做对比。该算法解决的问题是,给定一个不确定时间序列数据集,研究如何从该数据集中快速找出与查询序列 Q 距离最近的不确定时间序列。该方法是针对不确定时间序列上的老问题设计的新算法,其最大缺点是虽然设计了很多提高性能的优化技术,但时间开销依然很高。

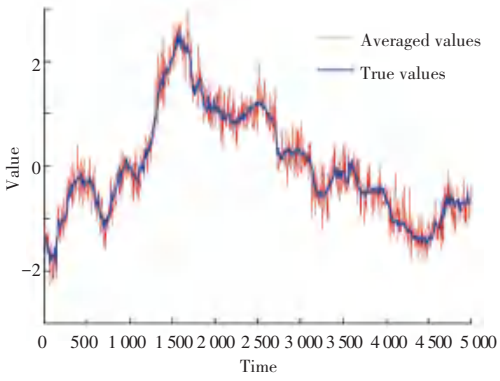
3.4 实验结果

首先从 UCR 数据集中随机选取了一条时间序列数据,在该条时间序列上生成带有噪声的多个样本,对这些样本求得的均值如图 1(a)所示,本文基于能量过滤后的时间序列如图 1(b)所示。2 个图中真实序列用粗实线表示,实验结果所得曲线用细线表示。从图中可以看出过滤后的曲线比直接求均值的曲线平滑很多,也更接近真实值。

其后在同一个时间序列上对比了过滤前后的时间序列与真实值之间的欧氏距离。如图 2 所示。

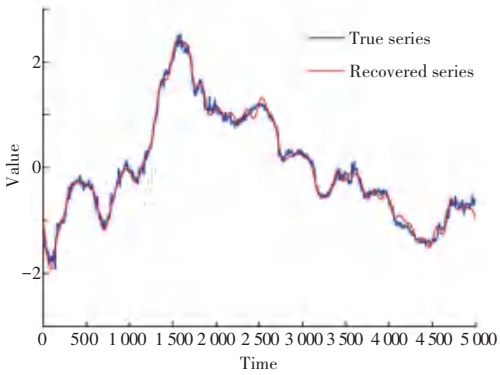
标识的曲线是过滤过程中最优过滤阈值所得到的结果,可以看出过滤后的曲线比过滤前的曲线更接近真实值,而在最优阈值频率和本文所选择的阈值频率所得的结果十分接近,几乎完全重合,说明本文所提的阈值频率估算方法是符合实际预期的。

接下来对比了不确定数据上 Top-k 运算的时间以及空间代价,如图 3 所示。注意到纵坐标是对数坐标系,可以看出和 Holistic-PkNN 相比,在利用本文方法清洗之后的确定序列上直接求 Top-k 的时间代价和空间代价都小很多。



（a）真实值与均值对比

（a）Comparison between real values and averaged values

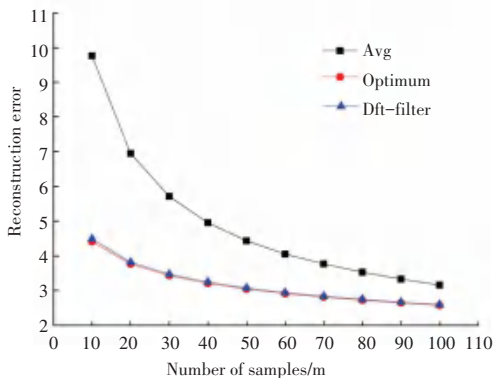


（b）真实值与过滤后的结果值对比

（b）Comparison between real values and filtered values

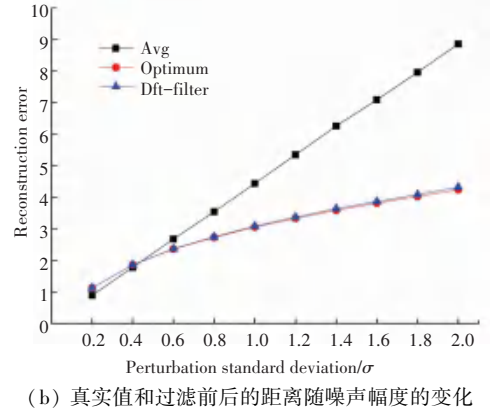
图 1　真实值和(a)求均值以及(b)过滤后的值之间的对比

Fig. 1　Comparison between real values and (a) averaged values and (b) filtered values



（a）真实值与过滤前后的距离随样本个数的变化

（a）Distance bwteen real values and (filtered) averaged values when varying number of samples

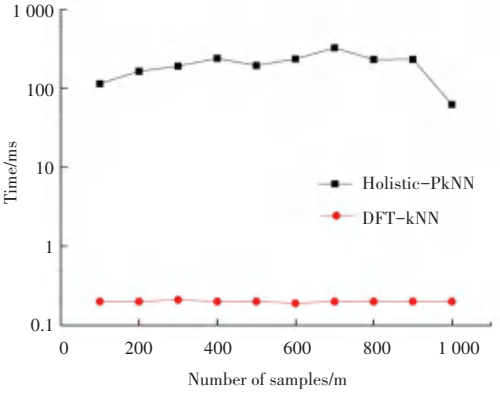


（b）真实值和过滤前后的距离随噪声幅度的变化

（b）Distance between real values and (filtered) averaged values when varying perturbation deviation

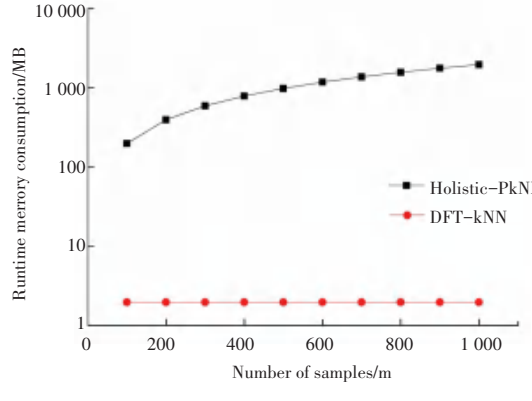
图 2　真实值和过滤前后的距离随(a)样本个数以及(b)噪声幅度的变化

Fig. 2　Distance between real values and (filtered) averaged values when varying (a) number of samples and (b) perturbation deviation



（a）不同 Top-k 算法的时间开销对比

（a）Comparision of different Top-k algorithms on time consumption



（b）不同 Top-k 算法的内存使用情况对比

（b）Comparison of different Top-k algorithms on memory usage

图 3　不同的 Top-k 算法的(a)时间开销以及(b)内存使用情况的对比

Fig. 3　Comparison of different Top-k algorithms on (a) time consumption and (b) memory usage

（下转第 12 页）