doi: 10.3969/j.issn.0490-6756.2019.04.009

双变换算法在多维序列数据分析中的优化研究

刘 云, 易 松
(昆明理工大学信息工程与自动化学院, 昆明 650500)

摘 要: 在流数据中,降低维度是处理多维序列数据的重要因素.提出一种双变换算法(DTA),针对在线序列数据,分别进行酉变换和双曲线旋转变换的双变换处理,得到假设函数的参数,通过牛顿算法迭代预测误差值,直到小于所预设的阈值,从而得到最优预测值.仿真结果表明,对比 OGD 和 RON 两种算法,DTA 算法在保障算法稳定性的前提下,有效减少计算时间.

关键词: 降维; 序列数据; 酉变换; 双曲线旋转变换
中图分类号: TN929.5 **文献标识码**: A **文章编号**: 0490-6756(2019)04-0633-06

Research on optimization of double transform algorithm in multidimensional sequence data analysis

LIU Yun, *YI Song*
(Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China)

Abstract: In streaming data, dimension reduction is an important factor in processing multidimensional sequence data. In this paper, a double transform algorithm (DTA) is proposed. For the online sequence data, unitary transformation and hyperbolic rotation transformation is carried out respectively in DTA, and the parameters of the hypothesis function are obtained. The error values are predicted by the Newton iterative algorithm, and the optimal prediction value is obtained until the error is less than the predefined threshold. The simulation results show that, compared with the two algorithms of OGD and RON, the DTA algorithm effectively reduces the computation time under the premise of ensuring the algorithm stability.

Keywords: Dimensionality reduction; Sequence data; Unitary transformation; Hyperbolic rotation transformation

1 引 言

随着云计算在互联网应用中的快速发展,序列数据的数据量大幅增加[1, 2].直接处理这些数据将面临"维数灾难","算法失效"等困难[3],因此迫切需要一种数据处理算法有效降低计算成本.

为了缓解计算成本过高的问题,许多学者提出使用在线处理算法,数据既不存储也不被重复使用,有效避免了批处理.Bedi 等人提出了一阶在线梯度下降算法(OGD ,the First Order Online Gradient Descent)[4, 5], OGD 算法仅使用梯度信息最小化代价函数,与其他在线学习算法相比,基于梯

收稿日期: 2018-04-23
基金项目: 国家自然科学基金(61262040)
作者简介: 刘云(1973—), 男, 副教授, 研究方向为人工智能研究. E-mail:liuyun@kmust.edu.cn
通讯作者: 易松. E-mail: 1090037776@qq.com

度的算法效率更高. 对于一个 M 维序列 $\{x_t\}_{t\geqslant 0}$, $x_t \in R$, OGD 算法的复杂度为 $O(M)$. 由于除了梯度信息以外,其他统计数据没有被使用,在达到最优解时,收敛速度十分缓慢. 在大多数大数据应用中,由于一阶学习算法的低计算成本被广泛使用,而使用二阶学习算法可以得到更加优异的性能. Mokhtari 等人提出了在线牛顿步(RON, the Regular Online Newton Descent Step)算法[6, 7], RON 算法由于采用代价函数的 Hessian 矩阵,可以提供更好的预测效果. 与一阶算法相比由于二阶算法采用二阶统计量,具有鲁棒性. 在收敛速度和稳定性上,RON 算法明显优于 OGD 算法. 但对于一个 M 维序列 $\{x_t\}_{t\geqslant 0}$, $x_t \in R$, RON 算法复杂度为 $O(M^2)$,而 OGD 算法复杂度为 $O(M)$,在实际大数据应用中通常采用复杂度较低的 OGD 算法[8].

在保障算法稳定性的前提下,为了有效降低计算时间,本文提出了一种双变换算法(DTA, Double Transform Algorithm). 首先对序列数据提取特征向量,分别进行酉变换和双曲线旋转变换,得到假设函数的参数. 然后通过牛顿算法迭代预测误差值,直到小于所预设的阈值,从而得到最优预测值. 该算法将传统二阶牛顿法的复杂度降低为 $O(M)$,并保留了牛顿法的收敛速度和稳定性.

2 模型构建

为了量化序列数据,用小写字母表示列向量,用大写字母表示矩阵. x^T 表示向量 x 的转置向量, I_M 表示 $M\times M$ 维单位矩阵, 0_M 表示 M 维零矩阵.

首先观察一个序列 $\{x_t\}_{t\geqslant 0}$, $x_t \in R$. 在每个时间点 t,估计值为 $\hat{x}_{t+1}$,其中 $\hat{x}_{t+1}\in R$,可用线性模型表达.

$$\hat{x}_{t+1} = w_t{}^T x_t + c_t \tag{1}$$

其中, $x_t \in R^M$ 表示样本的前 M 个特征向量,即 $x_t = [x_t, x_{t-1}, \ldots, x_{t-M}]^T$. 权重向量和偏移量可分别表示: $w_t \in R^M$, $c_t \in R^M$. 将 w_t 与 c_t 合并表示为: $w_t = [w_t; c_t]$,则估计值为: $\hat{x}_{t+1} = w_t{}^T x_t$,其中 $x_t = [x_t; 1]$. 绝对损失误差作为代价函数,即: $l_t(w_t) = \| e_t \|$,其中每个时刻的预测误差由下式给出.

$$e_t = x_{t+1} - \hat{x}_{t+1} \tag{2}$$

通过训练使代价函数最小,并获得最优;权系数 $\hat{w}_n$,即

$$\hat{w}_n = \arg\min_{w\in R^M} \sum_{t=0}^{n} \| x_{t+1} - w^T x_t \| \tag{3}$$

当使用二阶在线牛顿步算法(RON)训练模型时, M 维权向量更新项可表达为

$$w_t = w_{t-1} - \frac{1}{\mu} A_t{}^{-1} \nabla_t \tag{4}$$

其中, $\mu \in R$ 为步长; ∇_t 为代价函数 $l_t(w_t)$ 的梯度 $\nabla_t \overset{\Delta}{=\!=} \nabla l_t(w_t)$. 那么, $M\times M$ 维矩阵 A_t 可表示为

$$A_t = \sum_{i=0}^{t} \nabla_i \nabla_i{}^T + \alpha I_M \tag{5}$$

其中, $\alpha > 0$ 时, A_t 的顺序主子式均为正,可以保证 A_t 为正定矩阵. 由于 A_t 为正定矩阵,所以 $|A_t| > 0$, A_t 具有可逆性.

式(5)中定义的 A_t 具有递归结构,可表示为: $A_t = A_{t-1} + \nabla_t \nabla_t{}^T$,其中初始值为 $A_{-1} = \alpha I_M$. 因此对式(5)求逆,可以得到从 A_{t-1}^{-1} 到 A_t^{-1} 的更新,如下式.

$$A_t^{-1} = A_{t-1}^{-1} - \frac{A_{t-1}^{-1} \nabla_t \nabla_t{}^T A_{t-1}^{-1}}{1 + \nabla_t{}^T A_{t-1}^{-1} \nabla_t} \tag{6}$$

将式(6)的两端同时乘上 ∇_t 并代入式(4)中,可得

$$w_t = w_{t-1} - \frac{1}{\mu}\left[\frac{A_{t-1}^{-1}\nabla_t}{1+\nabla_t^T A_{t-1}^{-1}\nabla_t}\right] \tag{7}$$

由于式(6)每次更新需要计算一个 M 维向量 $x_t \in R^M$ 和一个 $M\times M$ 维矩阵,所以二阶在线牛顿步算法复杂度为 $O(M^2)$. 而一阶算法只需要乘法和加法运算,复杂度为 $O(M)$[9-11]. 虽然二阶算法与一阶算法相比,提供了更快的收敛速度和更好的稳定性能,但计算复杂性问题阻止了它们的应用.

3 优化算法

DTA 算法采用酉变换和双曲旋转变换来代替式(6)和式(7)更新权向量 w_t 和 Hessian 矩阵的逆,即 A_t^{-1}. 通过牛顿算法迭代预测误差值,直到小于所预设的阈值,从而得到最优预测值.

时间序列数据的特征向量 x_t 和 x_{t+1} 是强相关的,两个连续向量之间的关系可表示为

$$[x_{t+1}, x_t{}^T] = [x_{t+1}^T, x_{t-M+1}] \tag{8}$$

代价函数可以表示为

$$l_t(w_t) = \| x_{t+1} - w_t^T x_t \| \tag{9}$$

由式(6)和式(7)可知, w_t 和 A_t^{-1} 的更新规则为

$$w_t = w_{t-1} \pm \frac{1}{\mu}\left[\frac{A_{t-1}^{-1}x_t}{1+x_t^T A_{t-1}^{-1} x_t}\right] \tag{10}$$

$$A_t^{-1} = A_{t-1}^{-1} - \frac{A_{t-1}^{-1} x_t x_t^T A_{t-1}^{-1}}{1 + x_t^T A_{t-1}^{-1} x_t} \tag{11}$$

首先,定义式(10)中所给出的更新规则的标准化项

$$\eta_t = 1 + x_t^T A_{t-1}^{-1} x_t \tag{12}$$

那么，连续项 η_t 和 η_{t-1} 的差可表示为

$$\eta_t-\eta_{t-1}=x_t^TA_{t-1}^{-1}x_t-x_{t-1}^TA_{t-2}^{-1}x_{t-1} \tag{13}$$

然后，定义一个 $(M+1)\times 1$ 维扩展向量 $\tilde{x}_t=[x_t,x_{t-1}^T]^T$，所以，式(13)等价于下式.

$$\eta_t-\eta_{t-1}=\tilde{x}_t{}^T\Delta_{t-1}\tilde{x}_t \tag{14}$$

其中，更新项 Δ_{t-1} 被定义为

$$\Delta_{t-1}\overset{\Delta}{=\!=}\begin{bmatrix}A_{t-1}^{-1} & 0_M\\ 0_M^T & 0\end{bmatrix}-\begin{bmatrix}0 & 0_M^T\\ 0_M & A_{t-2}^{-1}\end{bmatrix} \tag{15}$$

由式(15)可知，更新项可以只知道用于计算 η_t 而定义的差值 Δ_{t-1} 的值，而不需要 A_{t-1}^{-1} 和 A_{t-2}^{-1} 的准确值. 更新项通过二阶矩阵表示，这是降低算法复杂度的关键.

假设 $t<0$ 时 $x_t=0$，由式(5)可得 $A_{-1}^{-1}=A_{-2}^{-1}=\frac{1}{\alpha}I_M$. 所以 Δ_{-1} 可表示为

$$\Delta_{-1}=\frac{1}{\alpha}\mathrm{diag}\{1,0,\ldots,0,-1\} \tag{16}$$

定义一个 $(M+1)\times 2$ 维矩阵 Λ_{-1} 和一个 2×2 维矩阵 $\prod_{-1}$ 为

$$\Lambda_{-1}=\sqrt{\frac{1}{\alpha}}\begin{bmatrix}1 & 0 & \ldots & 0 & 0\\ 0 & 0 & \ldots & 0 & 1\end{bmatrix}^T,\ \Pi_{-1}=\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \tag{17}$$

则 Δ_{-1} 可由矩阵 Λ_{-1} 和 $\prod_{-1}$ 表示为

$$\Delta_{-1}=\Lambda_{-1}\Pi_{-1}\Lambda_{-1}^T \tag{18}$$

根据式(18)，式(14)中 η_t 项等价于下式.

$$\eta_t=\eta_{t-1}+\tilde{x}_t{}^T\Lambda_{t-1}\Pi_{t-1}\Lambda_{t-1}^T\tilde{x}_t \tag{19}$$

式(19)可用矩阵形式表示为

$$\begin{bmatrix}\sqrt{\eta_t} & 0_2^T\end{bmatrix}\begin{bmatrix}\sqrt{\eta_t}\\ 0_2\end{bmatrix}=\begin{bmatrix}\sqrt{\eta_{t-1}} & \tilde{x}_t{}^T\Lambda_{t-1}\end{bmatrix}\Theta_{t-1}\begin{bmatrix}\sqrt{\eta_{t-1}}\\ \Lambda_{t-1}^T\tilde{x}_t\end{bmatrix} \tag{20}$$

其中 Θ_{t-1} 被定义为

$$\Theta_{t-1}\overset{\Delta}{=\!=}\begin{bmatrix}1 & 0_2^T\\ 0_2 & \Pi_{t-1}\end{bmatrix} \tag{21}$$

首先，使用酉空间的 Givens 变换 $H_{G,t}$ 来消除矢量 $[\sqrt{\eta_{t-1}},\tilde{x}_t{}^T\Lambda_{t-1}]$ 中第二个元素，然后用酉空间的双曲线旋转 H_{HB} 来消除上一项，即，$H_{HB,t}\Theta_{t-1}H_{HB,t}^T=\Theta_{t-1}$. 因此，我们实现了以下更新规则.

$$\begin{bmatrix}\sqrt{\eta_t} & 0_2^T\end{bmatrix}=\begin{bmatrix}\sqrt{\eta_{t-1}} & \tilde{x}_t{}^T\Lambda_{t-1}\end{bmatrix}H_t \tag{22}$$

其中，H_t 表示整个转换过程，由式(22)可得到下一个标准化项 η_t.

由式(14)和式(15)可得

$$\begin{bmatrix}A_{t-1}^{-1}x_t\\ 0\end{bmatrix}=\begin{bmatrix}0\\ A_{t-2}^{-1}x_{t-1}\end{bmatrix}+\Delta_{t-1}\tilde{x}_t \tag{23}$$

通过 H_t 变换可得标准项 $\sqrt{\eta_t}$，也可用 H_t 将 Λ_{t-1} 变换为 Λ_t，$A_{t-2}^{-1}x_{t-1}$ 变换为 $A_{t-1}^{-1}x_t$，如果将变换后的矢量进行拓展，如下式.

$$\begin{bmatrix}\sqrt{\eta_{t-1}} & \tilde{x}_t{}^T\Lambda_{t-1}\\ \frac{1}{\sqrt{\eta_{t-1}}}\begin{bmatrix}0\\ A_{t-2}^{-1}x_{t-1}\end{bmatrix} & \Lambda_{t-1}\end{bmatrix}H_t=\begin{bmatrix}\sqrt{\eta_t} & 0_2^T\\ q & Q\end{bmatrix} \tag{24}$$

其中，$q=\frac{1}{\sqrt{\eta_t}}[x_t^TA_{t-1}^{-1},0]^T$ 且 $Q=\Lambda_t$. 若将式(24)表示为 $BH_t=\tilde{B}$，其中 B 表示输入矩阵，$\tilde{B}$ 表示变换后的输出矩阵. 那么，可得以下等式.

$$\tilde{B}\Theta_{t-1}\tilde{B}^T=B\Theta_{t-1}B^T \tag{25}$$

因为 H_t 为 Θ_{t-1} 的酉矩阵，即，$BH_t\Theta_{t-1}H_t^TB^T=B\Theta_{t-1}B^T$. 所以由式(25)可生成以下矩阵元素.

$$\begin{aligned}&q\sqrt{\eta_t}=\begin{bmatrix}0\\ A_{t-2}^{-1}x_{t-1}\end{bmatrix}+\Delta_{t-1}\tilde{x}_t,\\ &qq^T+Q\Pi_{t-1}Q^T=\\ &\quad\frac{1}{\eta_{t-1}}\begin{bmatrix}0\\ A_{t-2}^{-1}x_{t-1}\end{bmatrix}\begin{bmatrix}0\\ A_{t-2}^{-1}x_{t-1}\end{bmatrix}^T+\Delta_{t-1}\end{aligned} \tag{26}$$

由式(23)可得，式(26)中 $q\sqrt{\eta_t}$ 等于 $[x_t^TA_{t-1}^{-1},0]^T$，q 由下式给出

$$q=\frac{1}{\sqrt{\eta_t}}\begin{bmatrix}A_{t-1}^{-1}x_t\\ 0\end{bmatrix} \tag{27}$$

因此，我们使用(26)中的第二项来确定 Q 矩阵的值

$$\begin{aligned}Q\Pi_{t-1}Q^T=&\begin{bmatrix}0 & 0_M^T\\ 0_M & \frac{A_{t-2}^{-1}x_{t-1}x_{t-1}^TA_{t-2}^{-1}}{\eta_{t-1}}\end{bmatrix}+\\ &\left(\begin{bmatrix}A_{t-1}^{-1} & 0_M\\ 0_M^T & 0\end{bmatrix}-\begin{bmatrix}0 & 0_M\\ 0_M^T & A_{t-2}^{-1}\end{bmatrix}\right)-qq^T\end{aligned} \tag{28}$$

使用式(15)拓展 Δ_{t-1} 项，由于 $A_{t-2}^{-1}-A_{t-1}^{-1}$ 等于 $\frac{1}{\eta_{t-1}}A_{t-2}^{-1}x_{t-1}x_{t-1}^TA_{t-2}^{-1}$，将此等式与 q 值代入式(28)，得

$$\begin{aligned}Q\Pi_{t-1}Q^T=&\begin{bmatrix}A_t^{-1} & 0_M\\ 0_M^T & 0\end{bmatrix}-\begin{bmatrix}0 & 0_M\\ 0_M^T & A_{t-1}^{-1}\end{bmatrix}=\\ &\Delta_t=\Lambda_t\Pi_t\Lambda_t^T\end{aligned} \tag{29}$$

式(29)表示 Π_t 不随时间改变，即，$\Pi_{t-1}=\Pi_t$ 且 Q 可表示为

$$Q=\Lambda_t \tag{30}$$

式(24)的变换提供了一个完整的更新规则，权重向量更新为

$$w_t=\begin{cases}w_{t-1}+\mathrm{sgn}(e_t)\frac{1}{\mu}\begin{bmatrix}\frac{A_{t-1}^{-1}x_t}{\eta_t}\end{bmatrix}, & \text{if } \|e_t\|>\varepsilon\\ w_{t-1}, & \text{otherwise}\end{cases} \tag{31}$$

由于经过变换后的矩阵 $\widetilde{B}$ 的维度为 $(M+2)\times 3$,所以 DTA 算法的复杂度为 $O(M)$. DTA 算法没有对序列数据提出任何统计假设,只对权向量的更新方法进行改进,所以与 RON 算法具有相同的均方误差.

FON 算法的伪代码具体如下,包含了所有的初始化和更新.

Data: $\{x_t\}_{t\geqslant 0}$ sequence

1) Choose $\alpha>0$, window size M and the step size μ;

2) $\Lambda_{-1}=\sqrt{\dfrac{1}{\alpha}}\begin{bmatrix}1 & 0 & \cdots & 0 & 0\\ 0 & 0 & \cdots & 0 & 1\end{bmatrix}^{\mathrm{T}}$;

3) $\amalg=\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$, $\Theta=\begin{bmatrix}1 & 0_2^T\\ 0_2 & \amalg\end{bmatrix}$;

4) $x_0=0_M$, $w_0=0_M$, $\eta_{-1}=1$, $\rho_{-1}=0_M$;

5) while $t\geqslant 0$ do

6) $\widetilde{x}_t=[x_t, x_t^T]^T$;

7) $\hat{x}_{t+1}=w_t^T x_t$;

8) $e_t=x_{t+1}-\hat{x}_{t+1}$;

9) $B=\begin{bmatrix}\sqrt{\eta_{t-1}} & \widetilde{x}_t^{\ T}\Lambda_{t-1}\\ \begin{bmatrix}0\\ \rho_{t-1}\end{bmatrix} & \Lambda_{t-1}\end{bmatrix}$;

10) Determine a Givens rotation $H_{G,t}$ for B

11) $\overline{B}=BH_{G,t}$;

12) Determine a Hyperbolic rotation $H_{HB,t}$ for $\overline{B}$;

13) $\begin{bmatrix}\sqrt{\eta_t} & 0_2^T\\ \begin{bmatrix}\rho_t\\ 0\end{bmatrix} & \Lambda_t\end{bmatrix}=B\overline{H}_{HB,t}$;

14) if $\|e_t\|>\in$ then

15) $w_{t+1}=w_t+\mathrm{sgn}(e_t)\dfrac{1}{\mu}\left[\dfrac{\rho_t\sqrt{\eta_t}}{\eta_t}\right]$;

16) $x_t=[x_t, x_{t-1}, \cdots, x_{t-M+1}]^T$;

17) end;

---

4 仿真分析

为了验证本文所提算法的性能,本文将所提算法 DTA 与类似常规算法 OGD 和 RON 在 Matlab 中进行类比实验,试验中对算法计算时间和稳定性两个指标进行了分析. 实验数据库为一组用于机器学习的标准测试数据集 UCI KDD[12-15],曾被大量的数据处理算法用于仿真比较.

4.1 稳定性分析

通过分别仿真混沌序列和伪周期合成时间序列来分析算法的稳定性.

首先从 UCI KDD 数据集中获取一个 Henon 映射混沌序列,该序列由以下函数生成:

$$x_t=1-\alpha x_{t-1}^2+\beta x_{t-2} \tag{32}$$

当 $\alpha=1.4$, $\beta=0.5$ 时,式(32)呈现混沌结构.

如图 1 所示,使用三种算法分别处理 64 维 Henon 映射混沌并进行稳定性分析. 随着样本数的增加,三种算法的均方误差逐渐降低. DTA 算法与 RON 算法均方误差值一样,表明 DTA 算法仍然具有二阶算法优异的稳定性. 与 OGD 算法对比,DTA 算法的均方误差值更小,具有更好的稳定性. 仿真结果表明,对多维混沌数据进行处理,DTA 算法仍然能够保障很好的稳定性.

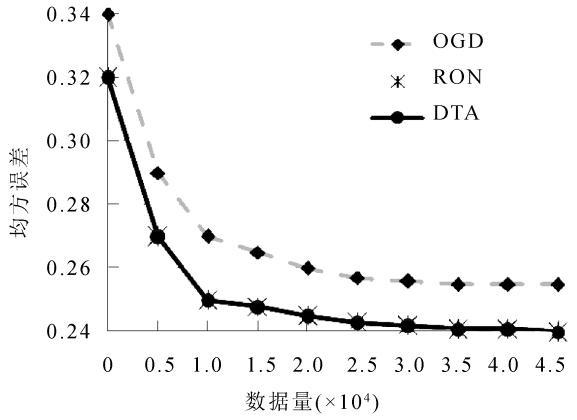

图 1 对 64 维混沌序列进行数值稳定性分析

Fig. 1 Numerical stability analysis of 64-dimensional chaotic sequences

然后从 UCI KDD 数据集中获取一个伪周期合成时间序列,该序列由以下函数生成.

$$\bar{y}=\sum_{i=3}^{7}\sin(2\pi(2^{2+i}+\mathrm{rand}(2^i))\tilde{t}) \tag{33}$$

其中,$\mathrm{rand}(2^i)$表示在一个均匀分布区间$(0, 2^i)$上产生一个随机值,矢量 $\tilde{t}$ 由均匀分布在区间$(0, 1)$上的 0.5 亿个样本组成,步长为 2×10^{-9}.

如图 2 所示,使用三种算法分别处理 64 维伪周期合成时间序列分析稳定性. 随着样本数的增加,DTA 算法与 RON 算法的均方误差逐渐降低并趋于平缓;而 OGD 算法的均方误差先增加再逐渐下降,并一直大于 DTA 算法的均方误差. 仿真结果表明,在处理多维伪周期合成时间序列时,DTA 算法具有更好的稳定性.

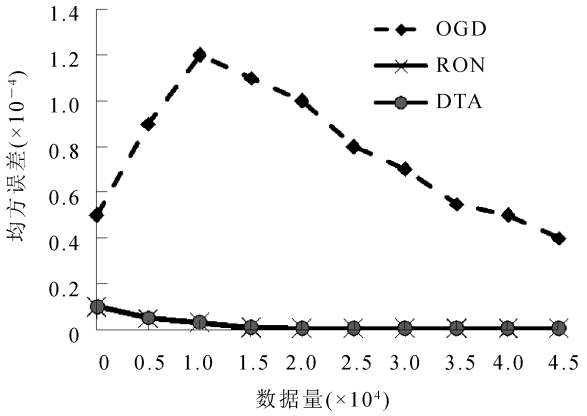

图 2　对 64 维伪周期合成时间序列数据序列进行稳定性分析

Fig. 2　Numerical stability analysis of 64-dimensional pseudo-cyclic synthesis time series data sequences

4.2　计算时间分析

分别对混沌序列和伪周期合成时间序列进行仿真，分析算法的计算时间.

图 3 表示在不同维度下，三种算法处理 5 亿个数据点的总计算时间对比. 随着维度的增加，三种算法的计算时间逐渐增加. 当样本数据有 20 个维度时，OGD 算法、DTA 算法与 RON 算法的计算时间分别为 1.6 h、2.1 h、2.2 h，即三种算法都能快速处理低维度大样本数据. 当样本数据有 100 个维度时，OGD 算法、DTA 算法与 RON 算法的计算时间分别为：1.9 h、2.5 h、14.2 h. 在处理高维度大样本数据时，OGD 算法与 DTA 算法计算时间短，依然可以快速处理数据，而 RON 算法计算时间长，是 DTA 算法的 5.7 倍. 仿真结果表明，DTA 算法有效减少多维混沌序列数据的计算时间.

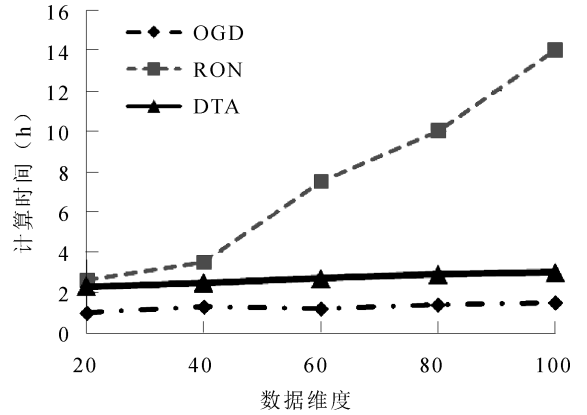

图 3　在不同特征维度下处理 5 亿个数据点总计算时间对比

Fig. 3　Comparison of the computation time with different feature dimension for processing 5 billion data points

图 4 表示三种算法处理 64 维伪周期合成时间序列数据达到稳定状态的总时间对比. DTA 算法达到稳定状态的时间比 OGD 算法降低 33.4%，比 RON 算法降低 75.6%. 对比 OGD 算法和 RON 算法，DTA 算法实现了较快的收敛速度，处理多维序列数据有明显优势.

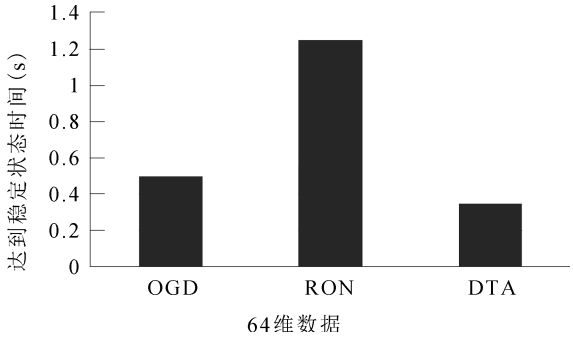

图 4　算法达到稳定状态的时间对比

Fig. 4　Total elapsed time that the algorithms reach the steady-state

通过对混沌序列和伪周期合成时间序列进行仿真，仿真结果表明，DTA 算法具有二阶算法高稳定性的优点，在保障稳定性的前提下，能有效降低计算时间.

5　结　论

选择有效的数据处理算法降低多维序列数据的计算成本是当前大数据应用的研究热点. 本文提出一种双变换算法(DTA)，针对在线序列数据，分别进行西变换和双曲线旋转变换的双变换处理，得到假设函数的参数，通过牛顿算法迭代预测误差值，直到小于所预设的阈值，从而得到最优预测值. 仿真结果表明，DTA 算法以一阶算法的低计算成本提供二阶算法的高性能，在保障算法稳定性的前提下，能有效降低计算时间. 在此后的研究工作中可以进一步优化预测精度并应用于商业中.

参考文献：

[1]　Meerkov S M, Ravichandran M T. Combating curse of dimensionality in resilient monitoring systems: conditions for lossless decomposition [J]. Trans Cybern, 2017, 47: 1263.

[2]　肖锦琦，王俊峰. 基于模糊哈希特征表示的恶意软件聚类方法 [J]. 四川大学学报：自然科学版，2018，55：476.

[3]　Kumari S, Jayaram B. Measuring concentration of distances-an effective and efficient empirical indexin

[J]. Trans Knowl Data En, 2017, 29: 373.

[4] Bedi A S, Sarma P, Rajawat K. Tracking moving agents via inexact online gradient descent algorithm [J]. IEEE J-STSP, 2018, 12: 202.

[5] Lai C, Feng G, Mukherjee K, *et al*. Maximum torque per ampere control for IPMSM using gradient descent algorithm based on measured speed harmonics [J]. Trans Ind Inform, 2018, 14: 1424.

[6] Mokhtari A, Ling Q, Ribeiro A. Network newton distributed optimization methods [J]. Trans Signal Process, 2017, 65: 146.

[7] 郭金刚, 董昊轩, 盛伟辉, 等. 电动汽车再生制能量回收最优控制策略 [J]. 江苏大学学报: 自然科学版, 2018, 39: 132.

[8] Abraham E, Stoianov I. Constraint-preconditioned inexact newton method for hydraulic simulation of large-scale water distribution networks [J]. Trans Control Netw Syst, 2017, 4: 610.

[9] Cohen K, Nedic A, Srikant R. On projected stochastic gradient descent algorithm with weighted averaging for least squares regression [J]. Trans Autom Control, 2017, 62: 5974.

[10] Bedi A S, Sarma P, Rajawat K. Tracking moving agents via inexact online gradient descent algorithm [J]. J-STSP, 2018, 12: 202.

[11] 易云飞, 林郭隆, 董文永. 基于维度近邻关系扩散的改进粒子群优化算法 [J]. 重庆邮电大学学报: 自然科学版, 2018, 30: 390.

[12] Jain A, Kanhangad V. Human activity classification in smartphones using accelerometer and gyroscope sensors [J]. Sens J, 2018, 18: 1169.

[13] Zhou Y, He J, Gu H. Partial label learning via gaussian processes [J]. Trans Cybernetics, 2017, 47: 4443.

[14] 刘富, 刘星, 康冰. 基于浅层残差网络的视线估计算法 [J]. 吉林大学学报: 信息科学版, 2018, 36: 333.

[15] 谭斌, 琚生根, 孙界平, 等. 基于状态转移的奖励值音乐推荐研究 [J]. 四川大学学报: 自然科学版, 2018, 55: 719.

引用本文格式:

中 文: 刘云, 易松. 双变换算法在多维序列数据分析中的优化研究 [J]. 四川大学学报: 自然科学版, 2019, 56: 633.

英 文: Liu Y, Yi S. Research on optimization of high dimensional sequence data analysis based on double-conversion preprocessing algorithm [J]. J Sichuan Univ: Nat Sci Ed, 2019, 56: 633.