基于概率模型的数据流预测查询算法*)

李国徽[1] 陈 辉[1] 杨 兵[1] 向 军[1,2] 陈 刚[1,3]
(华中科技大学计算机学院 武汉 430074)[1] (湖北民族学院信息工程学院 恩施 445000)[2]
(武汉科技大学计算机学院 武汉 430065)[3]

摘 要 挖掘在线数据流的变化趋势并预测未来时间窗口上的可能值,可以为许多时间敏感的应用提供重要决策支持。通过将数量可能无限的流数据元素映射到离散的且数量有限的流数据状态空间,不断变化的流数据变化趋势可以模拟成连续的流数据状态变化的过程,进而在很小的时间与空间代价下,数据流状态变迁的趋势动态存储在状态变迁图中。通过分析状态变迁图中的流数据变迁的统计规律,数据流上未来时刻的可能值可以应用马尔可夫模型在线连续预测。

关键词 数据流,流数据挖掘,数据预测,马尔可夫链

Predictive Queries Algorithm Based on Probability Model over Data Streams

LI Guo-Hui[1] CHEN Hui[1] YANG Bing[1] XIANG Jun[1,2] CHEN Gang[1,3]
(School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)[1]
(School of Information Engineering, Hubei Institute for Nationalities, Enshi 445000)[2]
(School of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065)[3]

Abstract Recently, data stream has widely appeared in many applications. Mining the evolving tendency and forecasting the data values in the future time windows of streams can provide important support for the future decision in many time-sensitive applications. For example, by using predictive queries in sensor networks for all kinds of monitoring, observers can forecast future values to detect abnormal events. By mapping the stream data into the stream state space, a continuously changing tendency of an online data stream can be modeled as a state transition process. After studying the history trend of the state transitions which are kept into state transition diagraph, the values in the next m time windows of a data stream can be forecasted efficiently. Extensive simulation experiments are conducted and show that the efficiency and precision of the proposed method are better than that of the existing analogous algorithms.

Keywords Data stream, Stream data mining, Data prediction, Markov chains

1 引言

近年来一类新的应用得到广泛关注,其中典型的例子主要包括:金融数据管理、网络监控、通信数据管理、Web 日志、传感器网络数据管理等等。在这些典型的应用中,数据的表现形式不是持久的关系,而是瞬态的数据流[1~5],数据的主要特征是大量数据高速连续到达,而不可能将数据全部保存到本地磁盘,然后再进行处理;对这类数据的处理都必须在内存中完成,并且所有的处理能在用户的响应时间要求内完成。

研究数据流的变化趋势,预测其在未来时刻的可能值,能够为数据流的应用提供重要决策支持。例如,在传感器网络系统中[1],传感器的数据输出模拟成为一个线性的随机系统,通过研究历史传感器数据的变化趋势,能够预测出其中可能丢失的值;在网格计算中[2],预测网络中的网络传输好坏,可以帮助用户选取合适的数据副本站点。

近年来,对时态数据进行趋势分析与数据预测已经取得了一定的研究成果。S. Papadimitriou 等在文[3]中提出了一种叫做 SPRINT 的算法来增量发现 n 个数据流中数据之间的关联关系,该方法能够挖掘这些数据流中能够反映数据流集合变化趋势的关键的隐藏变量。S. Papadimitriou 等还在文[5]中提出了一种发现时态数据序列中局部最佳模式的方法。D. Pokrajac 等在文[6]中提出了一种基于时空自回归模型的方法来分析历史数据上小样本并预测未来时刻的时空数据。K. Iwata 等在文[7]中提出了一种叫做 EPM 的预测模型,该模型基于多回归分析模型,并通过参数来保证数据预测的精度。为了提高回归预测模型的时间复杂度,A. Lazarevic 与 R. Kanapady 提出了一种局部聚类回归分析的方法来有效地进行灾难检测。在该方法中,首先在感兴趣的数据记录周围进行局部聚类,然后分析该簇范围的历史数据,对未来可能的灾难信息进行预测。

文[2,4,7,8]中提出的预测方法都是基于线性回归模型的,主要思想是使用曲线拟合的方法近似描绘数据的变化规律。这种方法在小样本的数据预测中效率很高,但是因为缺乏足够的训练样本而导致预测精度偏低。而在大样本的数据预测中,该方法却又很难获得准确的拟合曲线或者得到该曲线的计算代价会成倍增加。本文提出一种新的基于 Markov Chains 模型的数据流预测模型,*m*SetpForecast,来预测数据流上未来时刻的可能值。在该模型中,可能无限的流数据映射到有限的流数据状态空间中,同时数据流的变化趋势模拟成一个连续的状态变迁过程。随着时间的推移,数据流的历史状态变迁信息保存在状态变迁图(State Transition disGraph, STG)中,通过研究数据流上状态变迁的概率统计规律可以在线预测数据流在未来时刻的值。

2 模型设计与分析

本节从理论上分析数据流预测模型,并予以相关的理论

*)本文受青年国家自然科学基金(No. 60203017)、国家教育部博士点基金、湖北省杰出人才基金支持。**李国徽** 教授,博导,主要研究领域为主动、实时、移动数据库系统理论及集成技术;**陈 辉** 博士生,主要研究领域为时空数据库系统及数据挖掘;**杨 兵** 博士生,主要研究领域为移动数据库系统;**向 军** 博士生,主要研究领域为移动实时数据库服务质量;**陈 刚** 博士生,主要研究领域为时态数据库。

证明。尽管通过选择不同的映射函数，该模型可以适用于各种类型数据流的数据预测，但是为了讲述的方便，在本文中仅以实数据流为例进行讨论。

2.1 模型定义

数据流$\{X\}=\{X_i|X_i(data,timestamp),X_i\in R,i\in N\}$，$N$表示自然数，$R$表示实数。如果考虑流数据的所有可能值，则该可能值集的大小是无限的。如果仅仅观察$\{X\}$上最近的大小为$|W|$的滑动窗口[4]W内的数据样本，则可能值集的大小是有限的，记该可能值集为$D=\{d_i|d_i\in R,i\leqslant|W|\}$，$|D|$表示$D$的大小。假设在$\tau_i$时刻，变量$X$的值为$d_i(d_i\in D)$，在$\tau_j$时刻，$X$的值依概率变为$d_j(d_j\in D)$，则认为$X$的值由$d_i$变迁到$d_j$。任意时刻，变量$X$的值都有$|D|$种可能的变迁，它们称之为变量$X$的变迁趋势。

定义1(趋势T) τ_i时刻，变量X的趋势T_i可以记为$T_i=\{T_{ij}|T_{ij}(d_i,d_j,A_{ij}),i=1,2,\cdots,|D|\}$，表示$\tau_i$时刻变量$X$所有可能变迁的集合。其中$d_i$表示在$\tau_i$时刻变量的当前值，$d_j$表示在下一个时刻$\tau_j$时$X$的值，$A_{ij}$表示变量$X$从$d_i$变迁到$d_j$的概率，$|D|$表示变迁集合的个数，并且$\sum_{j=1}^{|D|}A_{ij}=1$。

对于数据流可能值集D中任意两个数据成员X_i与X_j，$X_i.data$与$X_j.data$的差异可能很小，以致于可以忽略。如果仍然严格区别并分别维护X_i与X_j的信息，那么需要庞大的存储空间来保存数据流变化趋势，代价太大。为了减小维护数据流变化趋势的代价，可以将可能无限的流数据元素映射到一个由若干值域宽度为k的子空间组成的状态空间中去。在每个子空间中，数据成员的取值很相近，可以近似用它们的均值来代替所有的数据成员。这样的一个子空间称之为数据流的一个状态。

定义2(状态S) 数据流上的一个状态定义为四元组$S(name,count,mean,time)$，其中$name$为状态S标识符，$count$为状态S的计数器，记录属于该状态的流数据元素的个数，$mean$为所有属于状态S的流数据元素的均值，$time$记录属于状态S的最近的一个流数据元素的时间戳。

数据流上的一个状态S_i可以表示为流数据元素的集合$S_i=\{X_{ir}|S_i.name-k/2\leqslant X_{ir}.data<S_i.name+k/2,r=1,2,\cdots,h\}$。假设$X_{i1},X_{i2},\cdots,X_{ih}$为属于状态$S_i$的$h$个数据成员，$p_{ir}$为数据成员$X_{ir}$在状态$S_i$中的概率，那么有

$$S_i.mean=\sum_{r=1}^{h}X_{ir}.data\times p_{ir} \tag{1}$$

定义3(状态变迁ST) 数据流上的一个状态变迁ST_{ij}定义为三元组$ST_{ij}(count,SStart,SEnd)$，表示从状态$SStart$到装态$SEnd$的变迁，其中$count$为状态变迁$ST_{ij}$的计数器，$SStart$表示起始状态，$SEnd$表示结束状态。

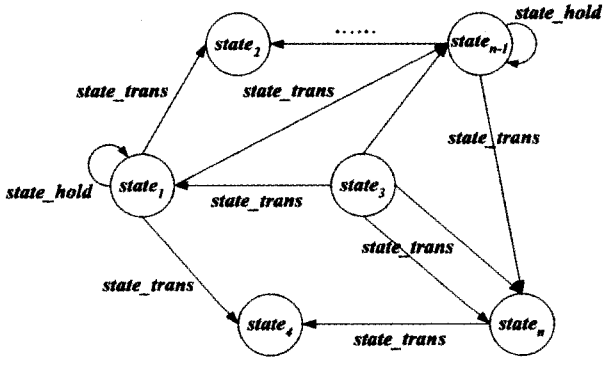

图1 STG模型

数据流上所有可能状态的集合称为数据流的状态空间(State Space,SSpace)，所有可能状态变迁的集合称为状态变迁空间(State Transition Space,STSpace)。如图1所示，状态变迁图(State Transition diaGraph,STG)用来维护数据流上状态变迁的历史，图中的顶点表示状态，弧表示状态变迁。如果用$\max(X.data)$表示流数据元素的最大值，$\min(X.data)$表示流数据元素的最小值，则状态空间的大小不大于$l(l\leqslant\lceil(\max(X.data)-\min(X.data))/k\rceil+1)$，而状态变迁空间的大小不大于$l^2$。显然，$l$远远小于数据流中可能值集的大小。

2.2 m步预测模型分析

假设数据流$\{X\}=\{X_1,X_2,\cdots,X_n,\cdots\}$，$n$表示数据流当前的大小。任意一个流数据元素$X_i$只可能属于$SSPace$中的某一个状态。假设$X_i\in S_j,S_j\in SSPace$，如果用$S_j.mean$代替$X_i$，则可以得到另外一个数据序列$\{Y\}$，且$\{Y\}$近似反映$\{X\}$中数据变化的趋势，很显然，维护$\{Y\}$中数据变迁的代价远小于维护$\{X\}$中数据变迁的代价。特别地，如果k的值足够小以至于每一个状态中仅仅包含一个流数据元素，那么数据序列$\{Y\}$将退化成$\{X\}$。

定义4(m步状态变迁mStepST) 假设在τ_0时刻，数据流的状态为S_i，经过m次状态变迁后，即τ_0+m时刻，数据流的状态变为S_j，则认为发生了从S_i到S_j的m步状态变迁，记为$ST_{ij}^{(m)}$，状态变迁$ST_{ij}^{(m)}$的概率为$A_{ij}^{(m)}$。

假设在τ_0时刻，变量X的值为$X_i(X_i\in S_i)$，并且m步状态转换概率矩阵为$A^{(m)}$，那么在τ_0+m时刻，流数据的可能值X_{i+m}可以由下式得到：

$$\hat{X}_{i+m}=\sum_{j=1}^{l}(S_j.mean\times A_{ij}^{(m)}) \tag{2}$$

定理1 统计量$\hat{X}_{i+m}$是随机变量X_{i+m}的无偏估计量。

证明：根据无偏估计量的定义，有

$$\begin{aligned}E(\hat{X}_{i+m})&=E(\sum_{j=1}^{l}(\sum_{r=1}^{h}X_{jr}P'_{jr})A_{ij}^{(m)})\\&=\sum_{j=1}^{l}(\sum_{r=1}^{h}E(X_{jr})p'_{jr})A_{ij}^{(m)}\\&=\sum_{j=1}^{l}(\sum_{r=1}^{h}E(X)p'_{jr})A_{ij}^{(m)}=E(X)\end{aligned}$$

定理2 假设数据流样本大小为n，则如果$n\to\infty$，数据流的单步($m=1$)预测值$\hat{X}=\sum_{j=1}^{l}(\sum_{r=1}^{h}X_{rj}p'_{rj})\times A_{ij}$收敛于流数据的期望。

证明：假设当前时刻，数据流$\{X\}$状态空间$SSpace$的大小为l，对于任意一个流数据X_i，X_i只可能属于$SSpace$中的某一个状态。如果Y_i表示$SSpace$中某一个状态S_j的状态均值，并用Y_i代替$\{X\}$中所有属于状态S_j的流数据元素，则可以得到数据序列$\{Y\}$。假设状态S_j包含h个数据成员$X_{j1},X_{j2},\cdots,X_{jh}$。如果$p_{jr}$表示数据成员$X_{jr}$在状态$S_j$中的概率，则状态$S_j$的均值$Y_j=\sum_{j=1}^{h}X_{jr}\times p_{jr}$，而统计量$\hat{X}=\sum_{j=1}^{l}(\sum_{r=1}^{h}{}_{rj}p'X_{rj})\times A_{ij}$可重写为$\hat{X}=\sum_{j=1}^{l}Y_i\times A_{ij}$。

假设在相隔很近的两个时刻τ_1、τ_2($\tau_1<\tau_2$)，数据流的状态均为S_i。而数据流的状态由S_i变迁到其他状态的真实概率为$A_{i1},A_{i2},\cdots,A_{il}$，则在下一个时刻流数据最可能的值为$\hat{Y}=\sum_{j=1}^{l}Y_j\times A_{ij}$。假设在$\tau_1$、$\tau_2$时刻，从状态变迁图$STG$中获取的由状态$S_i$变迁到其它状态的概率分别记为$A'_{i1},A'_{i2},\cdots,A'_{il}$和$A''_{i1},A''_{i2},\cdots,A''_{il}$，则根据式(2)，可以分别计算出数据流在$\tau_1$、$\tau_2$时刻的预测值$\hat{X}'$($\hat{X}'=\sum_{j=1}^{l}Y_j\times A'_{ij}$)与$\hat{X}''$($\hat{X}''=\sum_{j=1}^{l}Y_j\times A''_{ij}$)。根据贝努利试验与古典概率理论，数据流状态变迁的统计概率A''比A'更接近其状态变迁的真实概率A。也就是说预测值$\hat{X}''$比$\hat{X}'$更加接近于$\hat{Y}$，即$|\hat{X}''-\hat{Y}|\leqslant|\hat{X}'-\hat{Y}|$。

特别地，当数据流状态的值域宽度足够大，以致于数据流的每一个状态最多仅包含一个流数据元素，则数据序列$\{Y\}$退化成$\{X\}$，则$\hat{Y}$也就表示为$\{X\}$的数据期望。

3 预测算法设计

本节详细讨论状态变迁图的动态维护过程及数据流 m 步数据预测算法 mStepPrediction 的实现方法。

3.1 STG:构造与维护

随着时间的推移,数据流上状态变迁的过程增量更新至 STG 上。为了高效动态维护数据流变迁趋势,可以归纳出以下四种针对 STG 的操作。

假设在 τ_0 时刻,数据流的状态为 S_i,到 τ_0+1 时刻,新的流数据 X_j 到达。动态维护 STG 的四种操作可以描述为:(1)查找,首先在 STG 中查找 X_j 所属的状态,如果 $X_j \in S_j$,那么当前的状态变迁 ST_{ij} 就能很方便地确定,因为状态 S_i 在上一次查找中已经确定。(2)添加,如果查找 X_j 所属的状态失败,那么就创建一个新的状态 S_j 使其包含 X_j,并把 S_j 添加到数据流的状态空间中去,同时当前的状态变迁 ST_{ij} 也添加到状态变迁空间中。(3)更新,查找 X_j 所属的状态 S_j 成功后,更新状态 S_j 与状态 ST_{ij} 变迁的计数器和时间戳来记录最新的流数据变化趋势。(4)删除,随着时间的推移,删除那些长期不再出现的流数据状态及其相关的状态变迁。

当流数据连续到达时,状态变迁图 STG 由算法 UpdatingSTG 动态维护,每隔 $|W|$ 流数据,执行算法 Delete 周期性删除那些长期不再出现的状态以及相关的状态变迁,以减少算法所需要的存储空间和计算复杂度。详细的算法如图 2 所示,其中算法 UpdatingSTG 的时间复杂度为 $O(l^2)$,算法 Delete 的时间复杂度为 $O(l)$。

```
Procedure 1. UpdatingSTG( )
        Input:   A data stream DS
        Output: The State Transition diaGraph STG
        1  SSpace= ø; STSpace= ø; Si =NULL;
        2  for (each new data item Xi in DS){
        3  for (each state Sj in SSpace){
        4     if (Xi ∈Sj) {
        5        set Sj.timestamp = Xi.timestamp;
        6        set Sj.mean=( Sj.mean×Sj.count+Xi)/( Sj.count+1);
        7        set Sj.count+=1; } //end if of line 4
        8     else{
        9        create a new state Sj (Xi∈Sj) ;
        10       set Sj.timestamp=Xi.timestemp,
        11       set Sj.count=1, Sj.mean=Xi;
        12       insert Sj into SSpace;} // end else-if of line 8
        13    if(Si≠NULL){
        14    set STij.count+=1;//STij denotes a state transition
                      from Si to Sj;}
        15    else{
        16       set Si=Sj;}// end if-else-if of line 13-16
        17    }// end for of line 3
        18 }// end for of line 2
        19 return STG;

Procedure 2. Delete()
        1  for (each state Sj in SSpace){
        2  if(Xi.timestamp-Sj.timestamp>|W|){
        3     delete Sj;
        4     for(each state transition STuv in STSpace){
        5        if(Sv==Sj){
        6           set Su.count -=STuv.count;}
        7        delete STuv;}//end if of line 5
        8        }//end for of line 4
        9     }// end if of line 2
        10 }// end if of line 1
```

图 2 Updating STG 和 Delete 算法程序

3.2 m 步数据预测

假设 $A_1, A_2, \cdots, A_m$ ($m \ll |W|$)分别表示从 τ_1 时刻到 τ_m 时刻这 m 步状态变迁中每一次状态变迁的概率矩阵。根据全概率状态公式与马尔可夫链理论,m 步状态变迁的概率矩阵 $A^{(m)}$ 可以由下式得到:

$$A^{(m)} = \prod_{i=1}^{m} A_i \tag{3}$$

如果 $A_{1,ij}$ 表示在 τ_1 时刻从状态 S_i 到 S_j 的状态变迁概率,根据 STG 中的统计信息,$A_{1,ij}$ 可以由下式得到:

$$A_{1,ij} = \frac{ST_{ij}.count}{S_i.count} \tag{4}$$

如果 $A_{m,ij}$ ($1<m\ll|W|$)表示从状态 S_i 到 S_j 的 m 步状态变迁概率,当 $|W|\to\infty$ 且 $m\ll|W|$ 时,从状态 S_i 到 S_j 的状态变迁中,其单步状态变迁的概率 $A_{1,ij}$ 与 m 步状态变迁的概率 $A_{m,ij}$ 之间的差别很小,可以近似忽略,因此可以认为 $A_i = A_1$ ($1<i\leqslant m$),$A^{(m)} = \prod^m A_1$。在 τ_1 时刻,数据流上未来 m 个时间窗口内的值可以由算法 mStepForecast 预测。

```
Procedure 3. mStepForecast( )
        Input:    The current state of a data stream S'
        Output:   The prediction X'
        1  A=Ø; X'=0;
        2  for(each state Si in SSet){
        3     for(each state Sj in SSet){
        4        if(STij.count > 0){
        5           calculate the probability of STij Aij based on
                          formula (6);
        6        }//end if of line 4
        7     }//end for of line 3
        8  }//end for of line 2
        9  if(m>1)
        10    calculate A(m) based on formula (5);
        11 calculate the prediction X' based on formula (2);
        12 return X';
```

图 3 m Step Forecast 算法程序

任意时刻,根据状态变迁图 STG 计算数据流 m 步状态变迁概率矩阵 $A^{(m)}$ 的时间代价为 $O(\lceil \log_2^{m+1} \rceil \times l^2)$,因此算法 mStepForecast 的时间代价为 $O(\lceil \log_2^{m+1} \rceil \times l^3)$。假设 Δt 表示两个流数据元素到达的时间间隔,ξ 表示单个乘法运算的计算时间,则 m 步数据预测所需要的时间为 $\lceil \log_2^{m+1} \rceil \times l^3 \times \xi$。如果 $\lceil \log_2^{m+1} \rceil \times l^3 \times \xi \leqslant \Delta t$,则数据预测可以在下一个数据到达之前完成。

4 性能分析与实验

本节将通过一系列的实验来验证 mStepForecast 算法的可行性和正确性。实验在 CPU 为 1.8GHZ,内存为 766MB 的 PC 机上运行,所使用的操作系统为 WinXP。实验中使用两组真实数据:(I) 数字设备公司与世界其它地区互联的广域网交换机上一个小时内所交换的数据包[9];(II) 在 TAO 和 PIRATA 中用 ATLAS 浮标矩阵测量得到的距离海面上方 3 米处的海洋空气温度[10]。

为了评价算法性能,实验引入了三个性能评价标准[4]:(a) 平均绝对误差(Mean of Absolute Error,MAE)。$MAE = \frac{1}{t}\sum_{i=1}^{t}|\hat{X}_i - X_i|$,(b) 平均相对误差(Mean of Relative Error,MRE)。$MRE = \frac{1}{t}\sum_{i=1}^{t}\left|\frac{\hat{X}_i - X_i}{X_i}\right|$,(c)数据预测成功的比率(Ratio)。如果定义数据预测成功为该次数据预测结果的相对误差小于 10%,那么 $Ratio = \frac{\sum_{i=1}^{t}\hat{X}_i \text{ 预测成功}}{t}$,这里 X_i 表示流数据的真值,$\hat{X}_i$ 表示 X_i 的预测值,t 表示预测的次数。

(1)首先,研究算法的预测精度。如图 4 与 5 所示,当测试数据集为 $DS2$ 时,随着 k 增加,预测的误差缓慢增加,当 $k<11$ 时,预测的平均绝对误差不大于 5%;另外,随 k 值的增加,预测的成功率缓慢减小。但是对于数据集 $DS1$,随 k 值的增加,预测的误差先减小后增加,当 $k=10$ 时,误差最小,预测成功率最高。整体而言,算法在数据集 $DS2$ 上运行的预测精

度要明显优于在数据集 *DS1* 上的预测精度。其原因在于，对于线性数据集，数据的变化具有很强的规律性，在一个相对短的时间内，数据取值的变化不大，当 k 值比较小时，则状态变化更能反映数据流数值变化的特征。然而对于非线性数据集，数据的变化几乎没有什么规律性，状态变迁的趋势很难精确表现流数据的变化趋势。在本实验中，相比较而言，当 $k=10$ 时，DS1 中流数据的状态变迁的趋势最接近流数据的变化趋势。

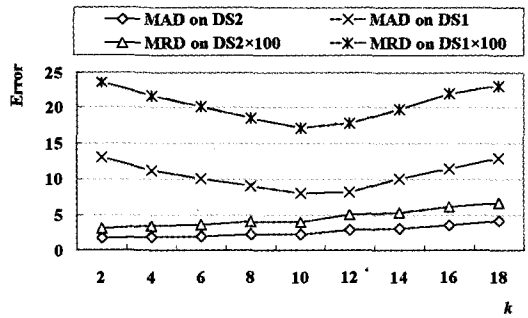

图 4　k 对 MAE 和 MRE 的影响

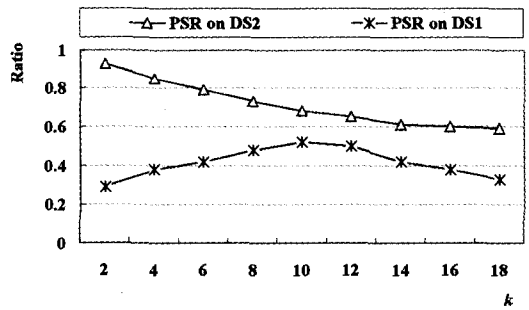

图 5　k 对 Ration 的影响

(2)第二，研究算法对未来 m 个时间窗口内的流数据进行预测时的精度。图 6 为在数据集 DS2 上进行多步预测的误差曲线。显然，随着 m 的增大，预测的误差逐步增大。其原因在于，在进行多步数据预测时，是根据当前的预测结果去预测下一步变迁的可能值。因此当 $m>1$ 时，由于预测误差的累积，多步预测中的误差将随着 m 的增加而迅速增大。

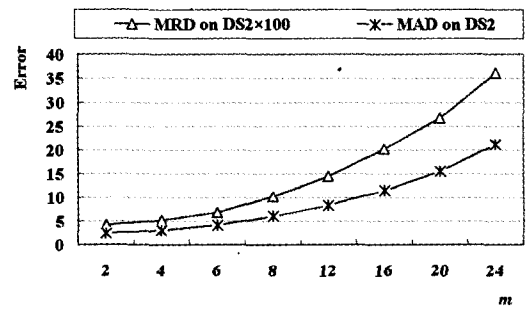

图 6　m 对 DS2 预测误差的影响

(3)最后，将 *mStepForecast* 算法与文[4]中提出的 *Multivariable Regression* 算法及文[7]中提出的 *Improved EPM* 算法进行性能对比分析。在对比实验中，使用的实验数据集为 *DS2*，*mStepForecast* 算法中的状态宽度 k 设置为 1，然后比较三种算法中的平均相对误差、平均绝对误差及预测成功率的最佳结果。文[4]与[7]中的算法是基于回归模型的，正如前文中分析的那样，这种方法在小样本的数据预测中效率较高，但是预测的精度较低，当数据样本比较大时，就很难获得一条比较准确的拟合曲线，并且获得拟合曲线的代价比较大。而 *mStepForecast* 算法是基于统计与概率模型的，当顺联样本集比较小时，误差相对较大，但随着流数据的增多，其数据的统计规律就越接近真实情况，预测的结果也越准确。另外，该方法计算复杂度与数据样本的大小无关。如图 7 所示，与文[4]与[7]中的方法比较起来，*mStepForecast* 方法具有较高的预测成功率和较小的预测误差。

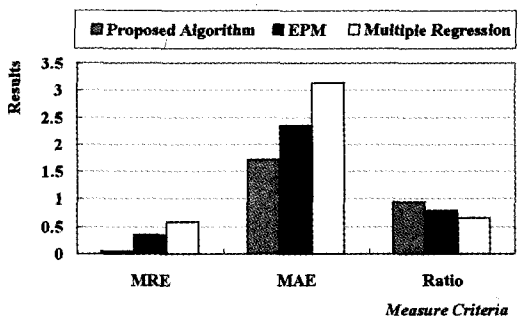

图 7　DS2 性能比较结果

结论　通过将可能无限的流数据元素映射到小得多的数据流状态空间中，连续到达的流数据的变化趋势模拟成连续的流数据状态变迁的过程。通过使用状态变迁图 *STG* 动态维护历史的数据流状态变迁过程，并研究流数据状态变迁的统计规律，数据流上未来时间窗口上的可能值可以使用马尔可夫链模型进行预测。仿真的结果显示，该方法能够高效地维护流数据动态变化的趋势，并对流数据未来时刻的可能值进行预测，并且相对于其它同类的算法，该算法的预测精度和预测成功率都较高。

参 考 文 献

1　Fletcher A K, Rangan S, Goyal V K. Estimation from lossy sensor data: jump linear modeling and Kalman filtering. In: Proceedings of the Third International Symposium on Information Processing in Sensor Networks, Berkeley, California, USA, April 2004

2　Vazhkudai S, Schopf J M. Using regression techniques to predict large data transfers. International Journal of High Performance Computing Applications, 2003, 17(3): 249～268

3　Papadimitriou S, Sun J, Faloutsos C. Streaming Pattern Discovery in Multiple Time-series. In: Proceedings of the 31st International Conference on Very Large Data Bases, VLDB 2005, Trondheim, Norway, August 30-September 2, 2005

4　李建中，郭龙江，张冬冬，等. 数据流上的预测聚集查询处理算法. 软件学报，2005，16(7)：1252～1261

5　Papadimitriou S, Yu P. Optimal multi-scale pattern in time series streams. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, Chicago, IL, USA, June 2006

6　Pokrajac D, Hoskinson R L, Obradovic Z. Modeling spatial-temporal data with a short observation history. Knowledge and Information Systems, 2003, 5(3): 368～386

7　Iwata K, Nakashima T, Anan Y, et al. Improving Accuracy of Multiple Regression Analysis for Effort Prediction Model. In: Proceedings of the 5PthP IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-based Software Engineering. Software Architecture and Reuse (ICIS-COMSAR06), July 2006

8　Lazarevic A, Kanapady R. Effective localized regression for damage detection in large complex mechanical structures. In: Proceedings of the Length ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Seattle, WA, USA, August 2004

9　Internet Traffic Archive, trace DEC-PKT-4. http://www.acm.org/sigcomm/ITA

10　Pacific Marine Environmental Laboratory. Topical atmosphere ocean project. http://www.pmel.noaa.gov/tao