RPAA:一种基于时间特性的时间序列建模表示

王元珍　李俊奎　曹忠升
(华中科技大学数据库与多媒体研究所　武汉 430074)

摘　要　滑动聚集平均近似 PAA(Piecewise Aggregate Approximation)是一种表示时间序列的方法,它通过时间序列上滑动一个等宽的滑动窗口将时间序列分成小的区段。考虑到时间序列的时间特性中不同区段的影响,本文提出了一种改进表示 RPAA(Reversed Piecewise Aggregate Approximation)。RPAA 表示对处于不同时间段的序列赋以不同的影响因子,具有线性时间复杂度,并且证明了 RPAA 满足下界定理,因而能够进行实际的查询。最后的实验表明该表示是有效的。

关键词　时间序列,表示,时间特性,影响因子

RPAA: A Time Property-based Representation for Time Series

WANG Yuan-Zhen　LI Jun-Kui　CAO Zhong-Sheng
(Research Institute of Database & Multimedia, Huazhong University of Science & Technology, Wuhan 430074)

Abstract　PAA (Piecewise Aggregate Approximation) is a method for representing time series by sliding an equal-width window on the data and dividing the data into segments. An improved representation method RPAA (Reversed Piecewise Aggregate Approximation) is presented by preserving the time property that different segments may play different role on the data in time series. RPAA assigns different influence factors to different segments and is of linear time complexity. It is proved that RPAA satisfies the lower-bounding lemma, so search on RPAA is practical. The experiments show the effectiveness of the method.

Keywords　*Time series, Representation, Time property, Influence factor*

1　引言

时间序列是一种多维的复杂数据类型,是由某个物理量在不同时间点的采样值按照时间先后次序排列而组成的序列,在科学、工程和商业领域有广泛应用。如:股票市场每天的股票收盘价格数据、每季度乘坐某次航班的旅客数、电话公司每小时的话务量等都是时间序列。近年来,对于时间序列数据的分析和挖掘激发了越来越多的研究人员的研究热情,而时间序列的建模表示则是其中的一个研究热点[4]。

Faloutsos 等人[3]提出了 GEMINI 框架来解决时间序列的建模方法。一般对于长度为 N 的时间序列可以看作是 N 维空间中的一个点。为了便于查找,可以首先利用 R tree, R* tree, k-d tree 等多维索引机制将这些点索引,然后查找则首先在索引上进行。不幸的是,由于目前现存的多维索引方式普遍仅对 8～10 维空间中的点比较有效,对于更高维的数据索引则将会导致索引性能的急剧下降,即引发所谓的"维度灾难"(Dimension Curse)问题[5],而实际的时间序列数据往往会远远超过这个界限。

为了解决维度灾难问题,一般的做法是对时间序列数据进行降维处理,然后再对降维后的数据进行索引。Agrawal R 等人[1]开创了使用离散傅立叶变换(Discrete Fourier Transform, DFT)进行时间序列数据降维的先河,先对原时间序列进行 DFT 操作,然后用前 K 个 DFT 表示的系数作为原时间序列的表示,其底层理论基础是数字信号处理领域中的 Parseval 能量定理。随后 Chan 等人[2]提出使用基于 Haar 小波变换的降维技术。Keogh 等人注意到大部分时间序列数据库中的数据可以先等分成区段,然后用区段中数据的均值来代表整个区段,基于此提出滑动聚集平均近似 PAA[5]的时间序列表示方法,并指出 PAA 表示方法较之 DFT 等频域表示方法具有简单易实现、易理解以及能够有效降维等优点,更重要的是它能够用于加权欧拉距离计算。

本文将时间特性引入到时间序列的表示中,提出一种 PAA 的改进表示方法,记为 RPAA 方法。这种表示的出发点是在很多情况下,对于不同时间区段的数据,其对于未来时间区段的数据的影响不同,距离当前时间近的数据影响较大。随着时间向后退,则影响越来越低,甚至可以忽略不计。因此,在计算时间序列间距离时应该将不同时间区段的数据影响区分开来。而 PAA 中对于不同区段的数据的平均表示,认为处于同等地位,在计算不同时间序列之间的距离时则相应地同等对待。

本文的其余部分如下组织:第 2 节讨论时间序列的 PAA 方法以及相似度度量;第 3 节是 RPAA 表示方法,以及相应的基于影响因子的相似度度量;我们的实验结果和对于实验结果的分析在第 4 节中给出,最后总结全文,并指出未来的进一步工作。

2　时间序列的 PAA 表示

2.1　相关定义

定义 1　时间序列(Time Series)　时间序列 $T=t_1, t_2, \cdots, t_n$ 是一串有序的 n 实数变量。

王元珍　教授,博士生导师,主要研究方向:现代数据库理论与实现技术、数据挖掘中间件技术;**李俊奎**　博士研究生,主要研究方向:数据挖掘、机器学习;**曹忠升**　副教授,主要研究方向:空间和多媒体数据库技术。

定义 2 时间序列长度(Length of Time Series) 对于有限长时间序列 $T=t_1,t_2,\cdots,t_n$，T 的长度为组成 T 的实数个数，记为$|T|$，即$|T|=n$。对于无限长时间序列，T 的长度定义为$|T|=\infty$。

无限长时间序列一般在数据流的建模中被使用，本文则主要讨论有限长时间序列。

定义 3 时间序列区段(子序列)(Segment, Subsequence) 给定长度为 n 的时间序列 T，T 的序列区段 C_s 是在 T 中从点 t_s 开始，数量为 $w(1\leqslant w\leqslant n)$ 个连续位置点所组成 T 的一个抽样，即

$$C_s=t_s,t_{s+1},\cdots,t_{s+w-1}$$

其中 $1\leqslant s\leqslant n-w+1$。

时间序列区段是通过在 T 上给定一个滑动窗口(窗口大小为 w)获得的。

定义 4 滑动窗口(Sliding Window) 给定一个长度为 n 的时间序列 T 和一个用户给定的区段长度 $w(1\leqslant w\leqslant n)$，$T$ 的所有子序列的矩阵 S 可以通过在 T 上滑动一个宽度为 w 的窗口，将每个区段 C_s 放入 S 的第 S 行得到，S 的大小为 $(n-w+1)*w$，w 为滑动窗口的大小。

2.2 PAA 表示方法

Keogh 等人[5]注意到对于任意的时间序列 T(长度为 n)，在其上滑动一个大小为 $w(w\ll n$，特殊情况下 $w=1)$ 的窗口，计算窗口中 w 个数据的均值，则所有均值按时间轴展开形成的序列是 T 的一个近似表示。基于此，他们提出了时间序列的 PAA 表示方法。

长度为 n 的时间序列 T 在 $N(1\leqslant N\leqslant n)$ 维空间中表示为向量 $\overline{T}_1,\overline{T}_2,\cdots,\overline{T}_N$，其中第 $i(1\leqslant i\leqslant N)$ 个元素 $\overline{T}_i$ 计算式为

$$\overline{T}_i=\frac{1}{w}\sum_{j=w(i-1)+1}^{wi}c_j \tag{1}$$

其中 $w=\lfloor n/N\rfloor$。

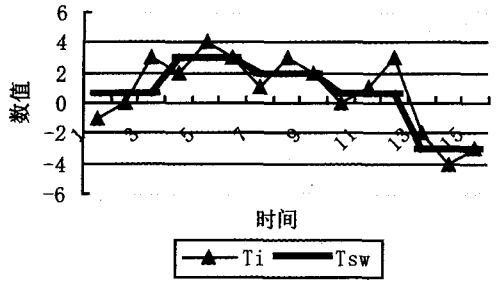
即为了将长度为 n 的时间序列降维，首先将其分 N 为个大小相同的帧，在每一帧中计算落入帧中数据的均值，然后将这些均值按照原时间轴排列组成长度为 N 的向量，以该向量作为原时间序列的表示。图 1 中给出了构造 PAA 表示的一个示例。

在等分中有两种极端情形：$N=n$ 时向量表示即为原时间序列；$N=1$ 时向量表示则是原时间序列的均值。

2.3 欧拉距离度量

欧拉距离度量在时间序列距离计算中被广泛使用。给定两个长度为 n 的时间序列 $T=t_1,t_2,\cdots,t_n$ 和 $C=c_1,c_2,\cdots,c_n$，则 T 和 C 之间的欧拉距离为：

$$D_{\text{euclidean}}(T,C)=\sqrt{\sum_{i=1}^{n}(t_i-c_i)^2} \tag{2}$$



$T=\langle -1,0,3,2,4,3,1,3,2,0,1,3,-2,-4,-3\rangle$

$\overline{T}=\langle 0.67,3,2,0.67,-3\rangle$ $n=15$ $N=5$

图 1 PAA 表示的索引值计算

为了防止因为递加数过多，而导致欧拉距离过大，可以计算加权的欧拉距离：

$$D_{\text{weuclidean}}(T,C)=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(t_i-c_i)^2} \tag{3}$$

由于函数 $f(x)=x^2$ 为单调函数，所以在实际计算中可以计算平方后的加权欧拉距离：

$$D_{\text{weuclidean\_square}}(T,C)=\frac{1}{n}\sum_{i=1}^{n}(t_i-c_i)^2 \tag{4}$$

PAA 表示支持对于平方后的加权欧拉距离的计算，将一个查询 C 也降维表示为

$$\overline{C}=\langle\overline{C}_1,\overline{C}_2,\cdots,\overline{C}_N\rangle$$

其中 $\overline{C}_i=\frac{1}{w}\sum_{j=w(i-1)+1}^{wi}c_j$，$w=[n/N]$ (5)

则此时计算 PAA 表示平方后的加权欧拉距离为

$$\begin{aligned}&D_{\text{weuclidean\_square PAA}}(T,C)\\&=\frac{1}{N}\sum_{i=1}^{N}(\overline{T}_i-\overline{C}_i)^2\\&=\frac{1}{N}\sum_{i=1}^{N}\Big[\frac{1}{w}\sum_{j=w(i-1)+1}^{wi}(t_j-c_j)\Big]^2\end{aligned} \tag{6}$$

其中 $w=[n/N]$。

3 时间序列的 RPAA 表示

3.1 表示方法来源

从上面对 PAA 表示方法的讨论中可以看出，实际计算两个时间序列的距离时所采用的策略是对时间序列分段平均，用对应区段均值求距离，又由(6)式，有

$$\begin{aligned}&D_{\text{weuclidean\_square\_PAA}}(T,C)\\&=\frac{1}{N}\sum_{i=1}^{N}(\overline{T}_i-\overline{C}_i)^2\\&=\sum_{i=1}^{N}\Big[\frac{1}{\sqrt{N}}(\overline{T}_i-\overline{C}_i)\Big]^2\end{aligned} \tag{7}$$

对于每个区段实际上采用的权值都为 $\frac{1}{\sqrt{N}}$，即对各区段实际是同等对待的。

但是在很多情况下，时间序列的演变过程中，对于时间序列数据的未来区段的影响是随着时间的前移而越来越小的，越接近时间序列当前时间的区段影响越大。其值对时间序列的预测的参考价值也越大。而远离的区段则影响较小，其值参考价值也较小，所以有必要对时间序列数据的这一特性建模。

3.2 RPAA 表示方法

考虑到时间序列区段间的不同影响，下面引入影响因子。

定义 5 影响因子(Influence Factor) 一个大于 0、小于 1 的实数，记为 ρ，$0<\rho<1$，表示两相邻时间区段中前一个时间区段对后一个时间区段施加的影响。

由于时间是一维的，具有不可逆转性，所以只有前序数据对于后继数据施加影响，反之并不成立。

定义 6 影响系数(Influence Coefficient) $Cof(T_i,T_j)$ 表示时间区段 T_i 对 T_j 的影响程度：

$$Cof(T_i,T_j)=\begin{cases}\rho^{j-i}T_i, & i\leqslant j\\0, & \text{其他}\end{cases} \tag{8}$$

PAA 中计算索引时将滑动窗口作用在原始时间序列 T 前端，然后沿着时间轴移动，计算各区段的均值，按照时间轴的方向组成索引向量。但是加入影响因子以后，这种按照时间轴方向的运算将必须每次都计算当前区段与最后区段的位

置距离$(j-i)$，这势必会导致ρ^{j-i}的多次重复计算。

为了解决这个问题，我们将滑动窗口首先作用在时间序列T尾端，然后沿着时间轴的逆方向移动，计算各区段的均值，以及相应的影响因子，从而形成索引向量，此即为时间序列的RPAA向量表示，此时

$$\bar{T}_i=\frac{1}{w}\rho^{i-1}\sum_{j=n-iw+1}^{n-(i-1)w}t_j,i=1,2,\cdots,N \tag{9}$$

其中$w=[n/N]$。

在图2中列出了构造时间序列的RPAA表示的算法。RPAA表示构造算法利用大小为w的滑动窗口沿时间轴逆向滑动，每滑动一次，计算均值和影响系数，从而计算出RPAA中该区段的索引量，由此可得RPAA的时间复杂度为$O(n)$。又RPAA需要额外的N维向量$\bar{T}$来保存中间的索引向量，所以RPAA的空间复杂度为$O(n)$。

3.3 距离计算

对于RPAA序列间的距离计算仍可以采用平方后的加权欧拉距离，此时有

$$\begin{aligned}&D_{\text{weuclidean\_ square RPAA}}(T,C)\\&=\frac{1}{N}\sum_{i=1}^{N}(\bar{T}_i-\bar{C}_i)^2\\&=\frac{1}{N}\sum_{i=1}^{N}[\frac{1}{w}\rho^{i-1}\sum_{j=n-iw+1}^{n-(i-1)w}(t_j-c_j)]^2\end{aligned} \tag{10}$$

其中$N=[n/w]$，$0<\rho<1$。

从(10)式可以看出，引入影响因子后，相邻两区段间前序区段对后序区段的影响计算为$\frac{\rho}{\sqrt{N}}$。随着时间的向前推进，靠近左端的区段对靠近右端的区段的影响越来越小$(0<\rho<1)$。

算法： 构造时间序列的RPAA表示
输入： $T=\langle t_1,t_2,\cdots,t_n\rangle$//长为$n$的时间序列
$w(1\leqslant w\leqslant n)$//滑动窗口大小
$\rho(o<\rho<1)$//时间序列间影响因子
输出： 时间序列T的RPAA表示
处理：
(1) $segment\_index=1$;//区段索引量
(2) $N=\lceil n/w\rceil$;//计算总的区段数(最后一//段数据量不足w则全部数据作为一区段)
(3) $\bar{T}=\varnothing$;//RPAA向量
(4) $factor=1$;//初始影响因子值
(5) **While**$(segment\_index<N)$**begin**
(6) 计算$\bar{T}_i=factor*\frac{1}{w}\sum_{j=n-iw+1}^{n-(i-1)w}t_j$;
//逆向滑动并计算索引$segment_index$向量
(7) 加$\bar{T}_i$到$\bar{T}$中;
(8) $factor=factor*\rho$;//影响因子值变小
(9) $segment\_index=segment\_index+1$;
(10) **End**;//while
(11) **Return**$\bar{T}$;//$\bar{T}=\langle\bar{T}_1,\bar{T}_2,\cdots,\bar{T}_N\rangle$

图2 RPAA表示构造算法

3.4 满足下界定理

对原始数据进行降维处理后，在索引空间的查找可能出现两类问题[7]。

• 漏查(False Dismissal)：在原始数据中两点距离小于给定的阈值ε，但是在索引空间中该两点距离却大于ε，从而对索引空间的点查询时发生漏查，即

$$\exists t_i,t_j\in T,\bar{t}_i,\bar{t}_j\in\bar{T},\varepsilon>0$$
$$D_{\text{index}}(\bar{t}_i,\bar{t}_j)\geqslant\varepsilon\Rightarrow d(t_i,t_j)<\varepsilon \tag{11}$$

• 错查(False Positive)：在索引空间中的两点距离小于给定的阈值ε，但是在原始数据中该两点距离却大于ε，从而对索引空间的点查询的结果中出现错查，即

$$\exists t_i,t_j\in T,\bar{t}_i,\bar{t}_j\in\bar{T},\varepsilon>0,$$
$$D_{\text{index}}(\bar{t}_i,\bar{t}_j)<\varepsilon\Rightarrow D(t_i,t_j)\geqslant\varepsilon \tag{12}$$

对于错查问题，可以通过针对索引空间中的查询结果再次到原始数据空间中查询，剔除其中$D(t_i,t_j)\geqslant\varepsilon$的点来解决。由于在索引空间中查询时已经剔除了大量不符合条件的点，只保留了原时间序列数据集合中一个很小的子集，所以再次在原始数据空间中查询时的耗费是可以接受的。漏查问题则决定了是否能够对时间序列进行有效的相似性查找，为了能够解决这个问题，Faloutsos[5]给出了降维下界定理(Lower Bounding)，即

$$D_{\text{index}}(T,C)\leqslant D(T,C) \tag{13}$$

下面证明RPAA满足下界定理。

定理1 RPAA满足欧拉距离的下界定理。

证明：由于$0<\rho<1$，所以

$$\begin{aligned}&D_{\text{weuclidean\_ square RPAA}}(T,C)\\&=\frac{1}{N}\sum_{i=1}^{N}[\frac{1}{w}\rho^{i-1}\sum_{j=n-iw+1}^{n-(i-1)w}(t_j-c_j)]^2\\&<\frac{1}{N}\sum_{i=1}^{N}[\frac{1}{w}\sum_{j=n-iw+1}^{n-(i-1)w}(t_j-c_j)]^2\\&=\frac{1}{N}\sum_{i=1}^{N}[\frac{1}{w}\sum_{j=w(i-1)+1}^{wi}(t_j-c_j)]\\&=D_{\text{weuclidean\_ square\_ PAA}}(T,C)\end{aligned} \tag{14}$$

所以$D_{\text{weuclidean_ square_ RPAA}}(T,C)$针对$D_{\text{weuclidean_ square_ PAA}}(T,C)$满足下界定理，而$D_{\text{weuclidean_ square_ PAA}}(T,C)$满足欧拉距离的下界定理[5]，所以得出$D_{\text{weuclidean_ square_ RPAA}}(T,C)$也满足欧拉距离的下界定理，从而定理得证。

4 实验研究

4.1 评价标准

这一节我们对RPAA的表示方法进行实验研究。RPAA与当前出现的时间序列表示不同的地方在于，它不仅考虑了时间序列的降维处理，而且考虑了时间序列不同区段间的相互影响。为了能够客观地评价RPAA的性能和效果，避免出现依赖于实现的结果[4]，必须首先给出实验的评价标准。

由于已经证明RPAA满足欧拉距离的下界定理，所以可以保证不会发生漏查问题。对于错查问题。引入评价标准：

$$F_{\text{rate}}=\frac{\text{错误报告记录数目}}{\text{查询的记录数目}}\times100\% \tag{15}$$

F_{rate}是每次查询中错误报告的比率，它只依赖于查询和数据。

4.2 实验环境

我们使用c++分别实现了PAA和RPAA时间序列表示。我们的实验环境是：Windows2000 Server，40G硬盘，256M内存，PIII-866 CPU，g++(mingw special)3.2.3编译环境。实验数据集选用UCI KDD Archive[6]的Synthetic Control Chart Time Series(SCCTS)和EEG。

SCCTS中包含600个合成的控制图的时间序列，每个时间序列长度为60，共有36000个时间数据，分为Normal，Cyclic，Increasing trend，Decreasing trend，Upward shift和Downward shift 6种类型。

EEG中包含了多个电极的脑电图时间序列，每个电极记录255个时间点数据。数据量为17M。

为方便实验处理，实验中首先用程序将EEG转化为单个电极的时间序列数据处于同一行。

4.3 实验结果

为了更好地描述数据特征，我们进行了多组实验，其中部分实验结果见表1和表2。

表1 $w=5,\rho=0.95,\varepsilon=33$ 的 F_{rate}

SCCTS	PAA	RPAA
Normal	0.67	0.54
Cyclic	0.53	0.47
Increasing Trend	0.60	0.23
Decreasing Trend	0.59	0.26
Upward Shift	0.51	0.44
Downward Shift	0.47	0.43

表2 $w=5,\rho=0.99,\varepsilon=15$ 的 F_{rate}

EEG	PAA	RPAA
A_1_co2a0000364	0.33	0.27
C_1_co2c0000337	0.28	0.15
A_m_co2a0000364	0.37	0.20
C_n_co2c0000337	0.31	0.14

4.4 实验分析

从表1和表2的实验结果中可以看出，RPAA较之PAA的结果有较大进步，能够进一步减少错查率。对于EEG的表示则能够比较好地拟合。

但PAA和RPAA均对SCCTS存在较大的错查率，这对数据量巨大的情形(如以G、T计算)，则是一种挑战。分析其原因是下界定理虽然保证了不漏查，但是在索引空间中的点距仍然比较大，所以在索引空间中不能够较大数量地剔除 $D(T,C)\leqslant\varepsilon$ 的记录。

结论 本文将时间特性引入到时间序列表示和距离度量中，提出了RPAA的表示方法。在时间序列相似性查找时兼顾了时间序列的整体特征和随时间变化的特性，进一步缩小了索引空间中点对距离，在查询过程中能够有效地减少查错率。

但在实验中我们发现，中间结果中错查率还有进一步缩小的空间，初步分析是PAA和RPAA虽然满足了时间序列降维的下界定理，保证无漏查，但是在索引比较的过程中的距离计算仍有缩小的余地。下一步工作将进一步研究时间序列降维下界定理，争取给出更加严格的时间序列降维下界定理，以缩小索引空间与实际数据空间的差距。

参考文献

1 Agrawal R, Faloutsos C, Swami A. Efficient Similarity search in sequence databases. In: Lomet D, ed. Proceedings of the 4th Conference on Foundations of Data Organization and Algorithms. Chicago, Illinois: Springer Verlag, 1993. 69～84

2 Chan K, Fu W. Efficient Time Series Matching by Wavelets. In: Proceedings of the 15th IEEE International Conference on Data Engineering. Sydney, Australia, 1999. 126～133

3 Faloutsos C, Ranganathan M, Manolopoulos Y. Fast subsequence matching in time-series databases. In: Proceedings of ACM SIGMOD Conference. Minneapolis, 1994. 419～429

4 Keogh E, Kasetty S. On the need for time series data mining benchmarks: a survey and empirical demonstration. In: the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Edmonton, Canada, 2002. 102～111

5 Keogh E, Pazzani M. A simple dimensionality reduction technique for fast similarity search in large time series databases. In: 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining. Kyoto, 2000

6 UCI KDD Archive. http://kdd.ics.uci.edu/, 2006

7 Zhu Yunyue. High Performance Data Mining in Time Series: Techniques and Case Studies: [PHD. Thesis]. New York City: N. Y. University, 2004. 138～139

(上接第55页)

定理1 设B树的高为 h，n 是外部节点的个数，且 $d=\lceil m/2\rceil$($\lceil m/2\rceil$ 指对 $m/2$ 取整)，则有：

$d^{h-1}+1\leqslant n\leqslant m^h$ 且 $\log_m n\leqslant h\leqslant\log_d(n-1)+1$。

证明：在树的第一层有1个节点，第二层有至多 m^1 个节点，第三层有至多 m^2……第 x 层有至多 m^x 个节点，因此该树至多有 m^h 个外部节点。在树的第一层有1个节点，第二层有至少 $d^{2-1}+1$ 个节点，第三层有至少 $d^{3-1}+1$……第 $x-1$ 层有至少 $d^{x-2}+1$ 个节点，第 x 层有至少 $d^{x-1}+1$ 个节点。因此有 $d^{h-1}+1\leqslant n\leqslant m^h$ 成立，由 $d^{h-1}+1\leqslant n\leqslant m^h$ 直接得到 $\log_m n\leqslant h\leqslant\log_d(n-1)+1$。

定理2 设 T 是一棵高度为 h 的 m 序B树，$d=\lceil m/2\rceil$ 且 n 是 T 中的节点个数，则 $2d^{h-1}\leqslant n\leqslant m^h$ 且 $\log_m n\leqslant h\leqslant\log_d(n/2)+1$。

证明：n 的上限源于 T 是一棵 m 叉搜索树。对于下限，注意相应的扩充B-树的外部节点都在 $h+1$ 层，而1,2,3,4,…,$h+1$ 层的节点最小数目是 $1,2,2d,2d^2,\cdots,2d^{h-1}$，因此B-树中外部节点的最小数是 $2d^{h-1}$。由于外部节点的数量比元素的个数多1，因此 $2d^{h-1}\leqslant n$，由 $2d^{h-1}\leqslant n\leqslant m^h$ 可直接得到 $\log_m n\leqslant h\leqslant\log_d(n/2)+1$。

成员加入且密钥更新开销最大情况下，如果加入点在第 r 层，那么密钥更新量是组播 $r+1$ 次，单播 h 次，总数是 $c_{join}=r+1+h$。因 $r\leqslant h-1$，所以：$c_{join}\leqslant 2h$。令 c_{join_max1} 和 c_{join_max2} 是最差情况下，密钥管理树分别为 m 序B树和B-树时，成员加入组播的最大密钥更新开销。有：$c_{join\_max1}\leqslant 2h_1$ 和 $c_{join\_max2}\leqslant 2h_2$，其中 h_1 和 h_2 分别是最差情况下，密钥管理树分别为 m 序B树和B-树时密钥树高。比较其开销量得：$c_{join\_max2}>c_{join\_max1}$。可见成员加入且组播密钥更新开销最大(即最差)情况下，本方案具有更小的密钥开销。

结束语 本文提出了一种利用m序B树实现密钥管理的方法。该方法的显著优点是新成员加入时在最差状况下的密钥更新开销比传统方案更小，提高了组播密钥更新效率。

参考文献

1 Cain B, Deering S, Kouvelas I, et al. Internet group management protocol [S]. Version 3. RFC3376, 2002

2 Wong Chung Kei, Gouda M, Lam S S. Secure group communications using key graphs [J]. IEEE/ ACM Trans. on Networking, 2000, 8(1): 16～30

3 Steiner M, Tsudik G, Waidner. Diffie-Hellman key distribution extended to group communications. In: Proc. the 3rd ACM Conference on Computer and Communications Security, New Delhi, India, 1996. 31～37

4 SteinerM, Tsudik G, Waidner. CLIQUES: A new approach to group key agreement. In: Proc. 18th IEEE International Conference on Distributed Computing Systems, Amsterdam, Netherlands, 1998. 380～387

5 A teniese G, Chevassut D, Detal H. The design of a group key agreement A P I. In: Proc. DARPA Information Survivability Conference and Exposition, SC, USA, 2000. 115～126

6 Harney H, Muckenhirn C. Group key management protocol (GKMP) architecture [S]. RFC2094, 1997

7 Ballardie T. Scalable Multicast Key Distribution. RFC 1949, 1996

8 Caronni G, WaldvogelM, Sun Detal. Efficient security for large and dynamic groups. In: Proc. the 7th Work shop on Enabling Technologies, (WETICE'98), Stanford, California

9 Dinsmore P T, Balenson D M, Metal H. Policy based security management for large dynamic groups: An overview of the DCCM project. In: Proc. the DARPA Information Survivability Conference & Exposition, SC, USA, 2000. 64～73

10 Wallner D, Harder E, Agee R. Key management for multicast: issues and architectures [S]. RFC2627, 1999

11 Goshi J, Ladner R E. Algorithms for Dynamic Multicast Key Distribution Trees. In: Proc. ACM Symp. Principles of Distributed Computing (PODC 2003), 2003