一种支持时间序列数据的 CBR 检索算法

史忠植[1] ,尹 超[1,2] ,叶世伟[2]
(1. 中国科学院计算技术研究所 智能信息处理重点实验室 ,北京 100080 ;2. 中国科学院研究生院 信息科学与工程学院 ,北京 100039)

摘 要:探讨了如何为 CBR(基于范例的推理)增加对一种特殊的范例类型 ——时间序列数据的支持. 分析了基于谱分析的时间序列相似度比较算法不适用于 CBR 检索的缺点 ,并在此基础上设计了一种综合性能很好的 CBR 检索算法. 思路是把时间序列相似度比较转化成一个卷积问题 ,并用 DFT 来简化这个卷积的计算. 通过对这种 CBR 检索算法进行了深入的理论分析和认真的实验 ,结果证明 ,提出的算法是一个高效的算法. 在这个检索算法的基础上 ,CBR 就能够应用到时序数据的分析推理中 ,具有广阔的应用前景.

关键词:基于范例的推理;时间序列数据;相似度比较
中图分类号: TP399 **文献标识码**: A **文章编号**:1673-4785(2007)01-0040-05

A CBR algorithm supporting time series data

SHI Zhong-zhi[1] , YIN Chao[1,2] , YE Shi-wei[2]
(1. Key Laboratory of Intelligent Information Processing , Institute of Computing Technology , Chinese Academy of Sciences , Beijing 100080 , China ;2. School of Information Science and Engineering Graduate University of Chinese Academy of Sciences , Beijing 100039 , China)

Abstract : This paper focuses on the retrieval algorithms of a special kind of CBR system in which cases are composed of time-series data. We introduced the classical algorithm used for processing similarity queries on time series data. This algorithm is based on the fact that DFT preserves the Euclidean distance in the time or frequency domain , and only the first few elements of the frequency sequence are significant , so the retrieval process can only use these significant elements to compute similarity degree. However , this algorithm has several disadvantages limiting its usage in CBR retrieval , so a new algorithm is presented for using batch method to compute the similarity degree. It is based on the observation that the original problem can be transformed to a convolution problem , and the circular convolution can be computed more efficiently using FFT. Theoretical analysis and experiment result prove that this algorithm is efficient and robust. The algorithm presented in this paper furnishes the CBR with the ability to process cases consist of time-series data.

Keywords : case-based reasoning; time series data; similarity comparison

基于范例的推理(case-based reasoning ,CBR)是实现人工智能的一种重要方法 ,它是对人类思维过程的模仿. CBR 在如下情形下效果比较好:1) 知识的主要来源是经验 ,而不是理论;2) 解决方案是可重用的;3) 目标是求出可行解而非最优解. 在过去的几年年里 ,CBR 的研究者们已经逐渐开始注意到时间信息的重要性 ,很多情况下 ,感兴趣的不仅仅是独立的快照(snapshot) ,而是一段连续的片段(episode) ,甚至是对将来的预测. 举个例子 ,在诊断病人的时候 ,医生不仅要了解患者目前的症状 ,也要了解其病史. 医生对同样的症状最终可能会制订不同的诊疗方案. 最近这方面有代表性的研究包括文献[2 - 3 ,16 - 17]等.

文献[2 - 3]的研究都是建立在 Allen 在文献

收稿日期:2006-07-10.
基金项目:国家自然科学基金资助项目(60435010 , 90604017 , 60675010) ;国家"973"资助项目(2003CB317004).

[4]中提出的时态模型(temporal model)的基础上的.文献[2]中提出了一种形式化逻辑方法来描述时序信息,从时序模型的观点来看,这个工作特别有意义,因为该文建立的模型中融合了基于时间点的元素和基于时间间隔的元素.然而,作者却没有给出检索算法,而只是声称可以采用图相似性算法来进行相似度检索.文献[3]则是在一个具体的应用项目的背景下讨论了如何在 CBR 中应用 Allen 的模型,然而由于该应用的一些特殊约束,所以该文的解决方法并没有普遍适用性. Allen 的时态模型是一种基于间隔的(interval - based).由于组合爆炸的缘故,这种模型只能处理一些简单的时序关系,而且相似性检索的计算量很大,效果也并不十分好.

文献[16 - 17]引入的时态模型是基于时间序列数据的,时间序列的相似度比较是个比较成熟的领域,有大量方法可以利用,比如时间规整(DTW),符号化,分段线型化,特征抽取降维技术等等.也正是因为这个原因,这 2 篇文章中没有过多的涉及具体的检索算法,而是侧重于设计时间序列 CBR 的框架和范例的存储结构.然而经过分析,CBR 中的时间序列相似度比较具有很多特殊性,需要设计有针对性的检索算法.

文中选择采用的时序模型是时间序列数据,着眼点不在于 CBR 的框架和范例的存储结构,而在于针对 CBR 中时间序列数据的特点,利用计算机在数值计算上的强大能力,设计高效的 CBR 检索算法.文中的研究不仅参考了 CBR 等人工智能领域的研究成果,也很大程度上参考了时间序列数据领域的一些成果.

1　支持时间序列的 CBR 模型

在推广 CBR 时遇到的一个最大的阻碍就是难于构造有效的范例库,CBR 的应用首先要求从客户的历史数据中抽取有代表性的数据构造范例库,这个过程不可能完全由机器完成,必须有人工参与进来.所以要向 CBR 增加时间序列支持,首先考虑的就是方便范例库的构造.实际应用中,用户往往只能确定某一段时间内的时间序列数据有特殊的含义,但并不能确定代表这种特殊含义的精确模式,在这种情况下最妥当的方法就是把这一段连续时间序列整个作为一个范例存入范例库中.

文中使用的表示符号如下:范例序列为 C,其长度均大于某个下限,其元素表示为 $C[i]$.查询序列为 Q,其长度均小于某个上限,其元素可表示为 $Q[i]$(下文为了简洁起见有时表示为 x_t).设 $len(C) = m$, $len(Q) = n, m \geq n$.每个范例序列 C 有 $m - n + 1$ 个长度为 n 的子序列 S,其元素为 $S[i]$(下文为了简洁起见有时表示为 y_t).设范例库中有 N 个范例序列,它们的长度不一定相等.

基于 CBR 的应用需要,范例库中任意范例序列 C 的长度 m 均大于查询序列 Q 的长度 n.查询的过程中,需要用大小为 n 的滑动窗口在 C 上截取出子序列 S,计算 S 与查询序列 Q 的相似度 $\mathrm{Sim}(Q,S)$,相似度最大的几个 S 作为查询结果返回,这实际上是一个子序列匹配问题.

时间序列相似度模型对 CBR 的推理效果起着重要作用,常用的有基于欧氏距离的,基于时间归整(dynamic time warping)的,基于界标(landmark)的,基于最长公共子序列(longest common subsequence)的,基于分段线性化表示的,基于概率方法的,等等.

出于减少计算量,减少误警(false alarm)等考虑,文中使用了基于欧氏距离的相似度模型,即把时间序列看成向量,定义相似度与欧氏距离成反比,欧氏距离越小,则相似度越大.这也是时间序列数据处理中应用最广泛的相似度模型,但它与人类的认知模型有一定的差异,因为一些相似度很低(即欧氏距离很大)的序列对人类的感觉而言却是相似的.

$$E(Q,S) = \left(\sum_{t=0}^{n-1} (x_t - y_t)^2 \right)^{1/2}. \qquad (1)$$

$$\mathrm{Sim}(Q,S) = - E(Q,S).$$

计算相似度的时间复杂度为 $O(n)$,对每个范例序列需要计算 $m - n + 1$ 次相似度.如果直接计算的话,对于大小为 N 的范例库,一次查询需要进行 N 次时间复杂度为 $O(n \cdot (m - n + 1))$ 的比较.文中把这种直接计算的方法称作 naive 方法,并作为比较各种算法效率的基准.

2　基于谱分析的相似度检索算法

基于欧氏距离计算相似度的方法中,Agrawal 等人[5]提出的基于离散傅里叶变换(DFT)的方法效果比较好.离散傅立叶变换公式如下,x_t 是时域数据,X_F 是变换域数据.

$$X_F = \frac{1}{\sqrt{n}} \sum_{t=0}^{n-1} x_t \exp\left(- \frac{\mathrm{j}2\pi Ft}{n} \right), F = 0,1,\cdots,n-1, \qquad (2)$$

$$x_t = \frac{1}{\sqrt{n}} \sum_{F=0}^{n-1} X_F \exp\left(\frac{\mathrm{j}2\pi Ft}{n} \right), t = 0,1,\cdots,n-1.$$

式中:如果 x_t 是实数,则 $(X_F)\ (F > 0)$ 一定是复数,且 X_F 和 $X_{n-F}\ (F > 0)$ 一定互相共轭.模 $|X_F|$ 定义

为频谱(spectrum),$|X_F|^2$ 定义为功率谱(power spectrum). X_0 比较特殊,一个序列中每个元素加上相同的值,则只影响 X_0,而 X_F,$(0 < F < n)$ 保持不变. 自然界大多数信号经 DFT 变换后所得频谱都是非均匀的,低频对应的频谱比较大,其频谱(以股票价格为例,去除 X_0)表现如图 1 所示.

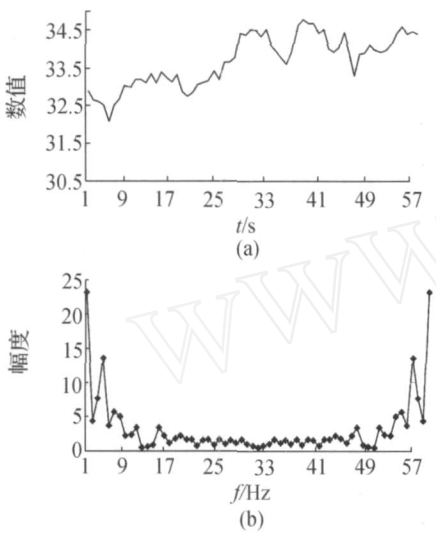


图 1 时间序列数据及其频谱

Fig. 1 Time-series data and its frequency spectrum

DFT 有如下 Parseval 定理:

$$\sum_{t=0}^{n-1}(x_t)^2 = \sum_{F=0}^{n-1}|X_F|^2. \quad (3)$$

同时 DFT 是一种线性变换,即有

$$[x_t + y_t] \Rightarrow [X_F - Y_F], \quad (4)$$

$$[ax_t] \Rightarrow [aX_F]. \quad (5)$$

式中:[ ]表示由式中:元素构成的向量,⇒表示傅里叶变换,由式(3)和式(4),可以推得

$$[x_t - y_t] \Rightarrow [X_F - Y_F]. \quad (6)$$

代入式(3)推得

$$E(Q,S) = E([X_F],[Y_F]) = \| XY \|_2 = \left(\sum_{F=0}^{n-1}|X_F - Y_F|^2\right)^{1/2}. \quad (7)$$

即时域数据的欧氏距离可以用傅里叶变换后所得频谱来求解.

由于频谱分布的不均匀性,可以只取频谱的 k 个主分量来计算欧氏距离. X_0 虽然很大,但由于它与序列的波形变化无关,应予以忽略. 计算 DFT 时可以略去公式,式(2)中系数 $\frac{1}{\sqrt{n}}$ 以优化计算,并不影响相似度的比较. 这样相似度定义为

$$E(Q,S) = E([X_F],[Y_F]) = \left(\sum_{F=0}^{k-1}(X_F - Y_F)^2\right)^{1/2}, 0 < k < n. \quad (8)$$

基于上述理论,文献[5-6]中采取的方法是事先计算出所有模式序列(对应于文中 CBR 中的范例)的谱变换系数,以若变换序列中若干低频项作为索引构建高维索引结构来进行相似度匹配.

3 时间序列 CBR 检索算法

3.1 已有算法的分析

这种方法效率很高,但是如果照搬到 CBR 系统中,会造成一些问题.

第一,缺乏灵活性. 结合 CBR 的具体应用,显然应当允许查询序列的长度可变,但是文献[5-6]中事先计算谱变换序列的方法,在变换之前要求固定序列的长度,这就限制了查询序列的长度,也就极大的限制了 CBR 的应用范围.

第二,文献[5-6]在谱变换系数中取 k 个主分量(即 $2k$ 个实数作为索引),建立维度为 $2k$ 的高维索引结构,可是经过研究,所有的高维索引结构,包括 R-tree 及其变种,pyramid-tree,空间填充曲线等,在高维情况下性能均退化到低于线性索引. 而且为了减少索引结构的规模,在建立高维索引过程中采取了很多近似手段,影响了查询精度,不适用于某些精度要求非常高的应用.

上述算法的核心思想是寻找时域运算在频域的等价计算方式,从中寻找优化的可能,在此思想指导下,提出了另一种利用 FFT 简化基于欧氏距离的相似度计算的改进方法.

3.2 改进方法

对每一条范例序列 C,用长度为 n 的滑动窗口截取 C 可得到 $m-n+1$ 个子序列,应用式(1)分别计算出 $m-n+1$ 个相似度,这个过程可表示成如下的形式. 为了推导的方便,用表示 Q 的元素,表示在 Q 与 S 比较过程中计算的第 t 个 $E(Q,S)$.

$$E(t)^2 = \sum_{t=0}^{n-1}(Q[i]^2 + \sum_{i=0}^{n-1} - C[t+i])^2, t = 0, \cdots, m-n. \quad (9)$$

因式分解得:

$$E(t)^2 = \sum_{i=0}^{n-1}(Q[i]^2 + \sum_{i=0}^{n-1}C[t+i]^2 - 2\sum_{i=0}^{n-1}Q[i]C[t+i], \quad t = 0, \cdots, m-n. \quad (10)$$

式中:$\sum_{i=0}^{n-1}Q[i]C(t+i)$ 是一个卷积,卷积的连续形式一般形如 $h(u) = \int_{-\infty}^{+\infty}f(x)g(u-x)\mathrm{d}x$,而卷积的离散形式一般是 $h(u) = \sum_{i=-\infty}^{+\infty}x[i]y[u-i]$.

为了把式(10) 中第 3 项化成卷积的标准形式,构造了序列 P,使得 $P[i] = Q(n - 1 - i)$,则式(9) 可以表示为

$$E[t]^2 = \sum_{i=0}^{n-1}(P[i] - C[u - i])^2,$$
$$u = n - 1, \cdots, m - 1. \quad (11)$$

因式分解得:

$$E(t)^2 = \sum_{i=0}^{n-1}P[i]^2 + \sum_{i=0}^{n-1}C[u - i]^2 -$$
$$2\sum_{i=0}^{n-1}P[i]C[u - i], u = n - 1, \cdots, m - 1. \quad (12)$$

式(12) 与式(10) 中的项是一一对应的,第 3 项单独列出如下:

$$H(u) = \sum_{i=0}^{n-1}P[i]C[u - i], u = n - 1, \cdots, m - 1. \quad (13)$$

可见 $H(u)$ 确实是一个离散卷积(亦称线性卷积). 直接计算 $H(u)$ 的时间复杂度为 $O(mn)$.

文献[15]中证明了一些有用的结论,第一,有限长序列的线性卷积等于循环卷积,而不产生混淆的必要条件是延拓周期 $L \geq N + M - 1$,式中 N、M 为 2 个有限长序列的长度. 第 2,傅里叶变换的循环卷积定理 ——时域的循环卷积对应于频域的乘积. 第 3,长度为 n 的离散序列的 FFT 可以在 $O(n\lg n)$ 内完成.

傅里叶变换的循环卷积定理的公式表达如下:

$$A \otimes B = i\mathrm{DFT}(\mathrm{DFT}(A) * \mathrm{DFT}(B)). \quad (14)$$

式中:⊗为循环卷积运算[12],文献[12]第 11 章有详细的说明和定义; * (点乘, componentwise product) 是这样一种运算:$\begin{bmatrix} a \\ b \end{bmatrix} * \begin{bmatrix} c \\ d \end{bmatrix} = \begin{bmatrix} ac \\ bd \end{bmatrix}$.

一个长度为 n 的序列和一个长度为 m 的序列卷积后的序列长度为 $m + n - 1$,即式(14) 中 $A \otimes B$ 的长度为 $m + n - 1$,所以 A、B 的长度也必须为 $m + n - 1$. 所以在对 P 和 C 作 FFT 之前,要先将 2 个序列末端补零转化为长度为 $m + n - 1$ 的 P 和 C!然后进行如下的计算过程:

$$P' \otimes Q' = i\mathrm{DFT}(\mathrm{DFT}(P') * \mathrm{DFT}(Q')). \quad (15)$$

由式(13) 得到的 $H(u)$ 序列的长度为 $m - n + 1$,而用式(15) 求得的 $P' \otimes C$ 长度为 $m + n - 1$,仔细分析 P 和 C 循环卷积的过程可知,$P' \otimes C$ 的头 $n - 1$ 项和尾 $n - 1$ 项不是需要的欧氏距离,应予以舍弃.

整个算法的分析如下:式(12) 中,对于一次查询而言,第 1 项在整个检索过程中只要计算一次,第 2 项可以事先计算并保存在范例库中,检索过程中无需重复计算,第 3 项即式(13) 中 $H(u)$, $u = n - 1$, …, $m - 1$ 的是主要的计算. 通过利用 FFT,计算 $H(u)$ 序列的时间复杂度从 $O(mn)$ 降低到 $O((m + n - 1)) \cdot \lg(m + n - 1))$. 这是一般情况下的性能,如果结合 CBR 的具体应用,还有进一步优化的可能:如果所有范例序列的长度 m 都一样,则对 P 补 $m - 1$ 个零得到的 P 是唯一的,每次查询过程只需对 P 作一次 FFT 计算,否则 N 次;如果限定查询序列长度 n 为固定大小,则 C 的每个补了 $n - 1$ 个零的子序列 S 也可事先确定,则每个 S 的 FFT 也可以事先计算并保存. 但是无论如何优化,每次查询都要计算 N 次频域的 componentwise 积和 N 次反变换,而计算第 3 项 $H(u)$, $u = n - 1$, …, $m - 1$ 的时间复杂度不会变,仍是 $O(m + n - 1) \cdot \lg(m + n - 1))$.

4　数值试验

从 Yahoo 财经网站上取得 100 支股票,每支截取10 000个数据点,构造范例库. 试验程序在 linux (内核版本 2. 6) 上用 C 实现,编译器是 GCC4. 0,傅立叶变换用的是 FFTW3 数学库.

根据数据的实际意义,有时需要先作一些预处理,比如对于股票价格序列,要调整数据以去除配股和分红对价格序列的影响,使得各个不同时期的数据具有可比性. 这点不予以详述.

一般希望图 1 中的频谱分布更加集中在少数几个主分量上,这就需要对原始数据进行滤波,滤除高频分量,在这里采用金融技术分析中常用的滑动平均法(Moving Averages):

$$\hat{x}_t = \frac{1}{w}\sum_{i=0}^{w-1}x_{t-i}.$$

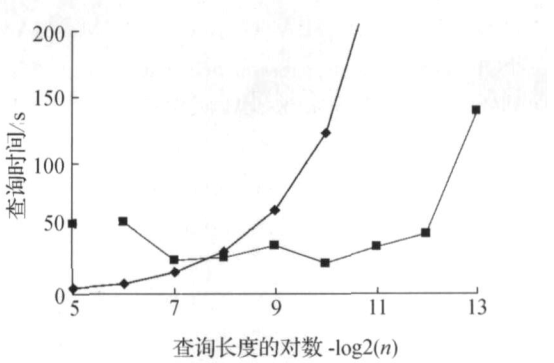


图 2　数值试验结果

Fig. 2　experiment result

试验结果如图所示. 纵坐标表示 10 次不同的查询所用时间,横坐标为查询序列长度(取对数坐标).

图中查询时间随范例序列长度 n 增长最快的是 naive 方法,随着查询序列长度的增长,改进方法性能优势越来越明显.试验结果验证了我们前文的分析,改进方法在时间复杂度上的表现非常好.

5 结束语

文中讨论了如何为 CBR 增加处理时间序列数据的能力,参考时间序列数据领域的研究成果,基于 CBR 的使用需要,我们设计了一种新的时间序列数据检索算法应用于 CBR.整个算法基于已有时间序列数据相似度匹配算法的核心思想,用卷积的方法批处理计算相似度,并用 FFT 简化卷积的计算过程,不仅保持了较低的时间复杂度,也增加了查询时的灵活性,方便了 CBR 的应用.我们的下一步工作是结合 CBR 在时间序列数据方面的具体应用,研究时间序列数据,对 CBR 的其他模块提出的新的要求,比如范例库维护等.

参考文献:

[1]AAMODT A,PLAZA E. Case-based reasoning: foundational issues, methodological variations, and system approaches[J]. AI Communications, 1994(7):56 - 72.

[2]MA J, KNIGHT B. A Framework for Historical Case - Based Reasoning[A]. ICCBR 2003[C]. Trondheim, Norway, 2003.

[3]JAERE M D, AAMODT A, SKALLE P. Representing temporal knowledge for case-based prediction[A]. ECCBR 2002[C]. Aberdeen, Scotland, UK, 2002.

[4]ALLEN J F. Maintaining knowledge about temporal intervals[J]. Communications of the ACM 1983, 26(11): 832 - 843.

[5]AGRAWAL R, FALOUTSOS C, SWAMI A. Efficient similarity search in sequence database[A]. FODO Conference[C]. Evanston, Illinois, 1993.

[6]FALOUTSOS C, RANGANATHAN M, MANOLOPOULOS Y. Fast subsequence matching in time-series database[A]. Proc of the ACM SIGMOD[C]. Minneapolis, Minnesota, 1994.

[7]GUTTMAN A. R-trees: a dynamic index structure for spatial searching[A]. Proceedings of the ACM SIGMOD [C]. Boston, MA, 1984.

[8]BERCHTOLD S., BOHM C, KRIEGEL H. The pyramid-technique: towards breaking the curse of dimensionality[A]. Proceedings of SIGMOD '98[C]. Seattle, Washington, USA, 1998.

[9]FALOUTSOS C, ROSEMAN S. Fractals for secondary key retrieval[R]. Technical Report UMIACS-TR-89-47, CS-TR-2242, University of Maryland, College Park, Maryland, 1989.

[10]BOHM C, BERCHTOLD S, KEIM D. Searching in high - dimensional spaces-index structures for improving the performance of multimedia databases[J]. ACM Computing Surveys, 2001, 33(3): 322 - 373.

[11]GAEDE V, GUNTHER O. Multidimensional access method[J]. ACM Computing Surveys, 1998, 30(2):221 - 290.

[12]BRACEWELL R. The fourier transform and its applications[M]. McGraw-Hill, 2000.

[13]史忠植.高级人工智能:第二版[M]. 北京:科学出版社, 2006.

[14]HAYKIN S, 叶世伟, 史忠植.神经网络原理[M].北京:机械工业出版社, 2004

[15]吴镇扬.数字信号处理[M].北京:高等教育出版社, 2004.

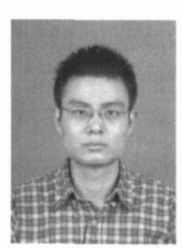
[16]MONTANI S, PORTINALE L. Case based representation and retrieval with time dependent features[A]. ICCBR 2005[C]. Chicago, IL, USA, 2005.

[17]SANCHEZ M M, CORTES U. An approach for temporal case-based reasoning: episode-based reasoning[A]. ICCBR 2005[C]. Chicago, IL, USA, 2005.

作者简介:

史忠植,男,1941 年生,中国科学院计算所主任研究员,博士生导师. IEEE 高级会员.主要研究领域为智能科学、分布智能、机器学习、知识工程等. 1979 年、1998 年、2001 年均获中国科学院科技进步二等奖,1994 年获中国科学院科技进步特等奖,2002 年获国家科技进步二等奖.

Email: shizz@ics.ict.ac.cn.



尹 超,男,1979 年生,硕士研究生,主要研究方向为智能信息处理,基于范例的推理技术.

E-mail: yinchao04@mails.gucas.ac.cn.

叶世伟 男,1968 年生,博士,副教授.中国科学院研究生院信息科学与工程学院;主要研方向为智能信息处理、神经计算与优化计算、认知科学等方面,在国内外重要刊物上发表论文 20 余篇.