TỔNG QUAN VỀ TÌM KIẾM TƯƠNG TỰ TRÊN DỮ LIỆU CHUỖI THỜI GIAN

Dương Tuấn Anh

Trường Đại học Bách Khoa, ĐHQG-HCM

(Bài nhận ngày 07 tháng 12 năm 2010, hoàn chỉnh sửa chữa ngày 20 tháng 04 năm 2011)

TÓM TẮT: *Dữ liệu chuỗi thời gian tồn tại trong nhiều ứng dụng thực tế, từ các lãnh vực khoa học kỹ thuật cho đến kinh tế, tài chính. Trong những ứng dụng này, việc tìm kiếm những chuỗi con truy vấn có xuất hiện trong cơ sở dữ liệu chuỗi thời gian là một công việc rất cần thiết. Sự truy tìm dựa vào độ tương tự như vậy là một mô đun căn bản trong nhiều công tác khai phá dữ liệu chuỗi thời gian cao cấp hơn như gom cụm, phân lớp, tìm mô típ, phát hiện mẫu bất thường, khám phá luật kết hợp và trực quan hóa dữ liệu. Mặc dù có nhiều cách tiếp cận khác nhau đã được đề xuất, hầu hết các cách tiếp cận đều dựa trên một tiền đề chung là các phương pháp thu giảm số chiều và các cấu trúc chỉ mục không gian. Bài tổng quan này điểm qua các nghiên cứu mới đây và cho thấy những phương pháp này hội tụ về một khung thức chung của sự rút trích đặc trưng.*

Từ khóa*: Chuỗi thời gian, tìm kiếm tương tự, thu giảm số chiều, rời rạc hóa, rút trích đặc trưng.*

1. GIỚI THIỆU

Một *chuỗi thời gian* (time series) là chuỗi trị số thực, mỗi trị biểu diễn một giá trị đo tại những thời điểm cách đều nhau. Những tập dữ liệu chuỗi thời gian rất lớn xuất hiện trong nhiều lãnh vực khác nhau như y khoa, kỹ thuật, kinh tế, tài chính, v.v…*Tìm kiếm tương tự* (similarity search) là công tác căn bản nhất để khai thác những cơ sở dữ liệu chuỗi thời gian.

Vài áp dụng của tìm kiếm tương tự như:

- nhận dạng những công ty có kiểu mẫu tăng trưởng giống nhau.
- Xác định những sản phẩm trong công ty có những kiểu mẫu doanh số bán hàng giống nhau.
- Xác định những chứng khoán có giá biến động theo một kiểu cách giống nhau.
- Tìm xem một giai điệu nhạc có tương tự với một đoạn nhạc nào trong tập hợp những bản nhạc đã có bản quyền.
- Tìm những tháng trong quá khứ mà lượng mưa giống như tháng vừa rồi.

Bài toán tìm kiếm tương tự nêu trên là một *thành phần căn bản* trong nhiều công tác khai phá dữ liệu chuỗi thời gian cao cấp hơn như gom cụm, phân lớp, tìm mô típ, phát hiện mẫu bất thường, khám phá luật kết hợp và trực quan hóa dữ liệu.

Bài viết tổng quan này nhằm mô tả một số tiến bộ gần đây của lãnh vực tìm kiếm tương tự trên dữ liệu chuỗi thời gian; những phương pháp cho phép truy vấn hữu hiệu những chuỗi con sử dụng những độ đo tương tự mềm dẻo để không bị ảnh hưởng bởi những phép biến đổi dữ liệu hoặc những sai sót dữ liệu. Bài tổng quan này sẽ cho thấy những phương pháp này

hội tụ về một dạng thức chung của sự rút trích đặc trưng (feature extraction).

2. BÀI TOÁN TÌM KIẾM TƯƠNG TỰ TRÊN DỮ LIỆU CHUỖI THỜI GIAN

Đối với bài toán tìm kiếm tương tự trên dữ liệu chuỗi thời gian thì dữ liệu được biểu diễn thành những dãy số thực, thí dụ $T = t_1,...t_n$. Cho hai chuỗi thời gian $X = x_1, x_2,...,x_n$ và $Y = y_1,y_2,...,y_n$. Ta cần phải tính độ tương tự *SIM(X, Y)* của hai chuỗi thời giann này.

2.1. Các độ đo tương tự

Đã có nhiều độ đo tương tự đã được sử dụng. Việc chọn một độ đo tương tự là tùy thuộc rất nhiều vào miền ứng dụng và trong nhiều trường hợp thì một độ đo thuộc chuẩn L_p đơn giản như độ đo Euclid là đủ tốt để dùng. Tuy nhiên trong nhiều trường hợp thì độ đo Euclid tỏ ra quá cứng nhắc vì không thích nghi được với những phép biến đổi như tịnh tiến (shifting), co giãn biên độ (scaling) hay xoắn trục thời gian (time warping). Nhiều phương pháp tìm kiếm tương tự mới hơn dựa vào những độ đo tương tự mềm dẻo và vững chắc hơn như độ đo xoắn thời gian động, chuỗi con chung dài nhất.

Độ đo Euclid

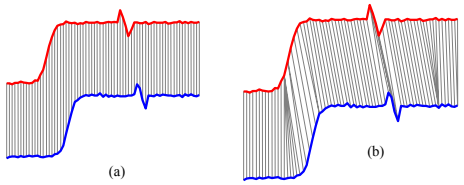
Cho hai chuỗi thời gian $Q = q_1...q_n$ và $C = c_1...c_n$ độ đo khoảng cách Euclid giữa hai chuỗi thời gian này được cho bởi công thức.

$$D(Q,C) \equiv \sqrt{\sum_{i=1}^{n}(q_i - c_i)^2}$$

Độ đo khoảng cách Euclid có ưu điểm là dễ hiểu, dễ tính toán, dễ mở rộng cho nhiều bài toán khai phá dữ liệu chuỗi thời gian khác như gom cụm, phân lớp, nhận dạng mô típ, v.v.. Nhưng độ đo khoảng cách này có nhược điểm là nhạy cảm với nhiễu và thiếu sự mềm dẻo khi so trùng.

Độ đo xoắn thời gian động

Việc so trùng 2 đường biểu diễn dữ liệu bằng cách tính khoảng cách từng cặp điểm 1-1 (điểm thứ *i* của đường thứ I so với điểm thứ *i* của đường thứ II) là không phù hợp trong trường hợp hai đường này không hoàn toàn giống nhau nhưng hình dạng biến đổi rất giống nhau. Như trong hình 1, hai đường biểu diễn rất giống nhau về hình dạng nhưng lệch nhau về thời gian. Trong trường hợp này, nếu tính khoảng cách bằng cách ánh xạ 1-1giữa 2 đường thì kết quả rất khác nhau và có thể dẫn đến kết quả cuối cùng không giống như mong muốn.



Hình 1.(a) Tính khoảng cách theo Euclid (b) tính khoảng cách theo *DWT*. (Từ nguồn [12])

Vì vậy để khắc phục nhược điểm này, thì một điểm có thể ánh xạ với nhiều điểm và ánh xạ này không thẳng hàng (xem hình 1b). Phương pháp này gọi là *xoắn thời gian động (*Dynamic Time Warping - DTW) được đề xuất bởi Bernt và Clifford, 1994. Chi tiết về cách tính DTW, độc giả có thể tham khảo thêm trong bài báo ([4]).

Phương pháp DTW có ưu điểm là cho kết quả chính xác hơn so với độ đo Euclid và cho phép nhận dạng mẫu có hình dạng giống nhau nhưng chiều dài hình dạng về thời gian có thể

khác nhau. Phương pháp này có nhược điểm là thời gian chạy lâu, tuy nhiên gần đây đã có những công trình tăng tốc độ tìm kiếm tương tự dùng độ đo DTW.

2.2. So trùng toàn bộ và so trùng chuỗi con

Mặc dù có nhiều loại khác nhau, nhưng các yêu cầu truy vấn trên dữ liệu chuỗi thời gian có thể chia làm 2 loại:

So trùng toàn bộ: (whole matching) Đối với những truy vấn so trùng toàn bộ thì chiều dài của chuỗi dữ liệu truy vấn và chiều dài chuỗi dữ liệu ban đầu là bằng nhau. Bài toán này ta thường được dùng trong việc gom cụm, hay phân loại dữ liệu chuỗi thời gian. Ví dụ, "tìm giá chứng khoán của những công ty nào thay đổi giống nhau".

So trùng chuỗi con:(subsequence matching) Trong trường hợp so trùng chuỗi con thì chiều dài của dữ liệu truy vấn ngắn hơn rất nhiều so với chiều dài của dữ liệu ban đầu. Vì vậy, nhiệm vụ chính là tìm những đoạn trong dữ liệu ban đầu tương tự với dữ liệu truy vấn. Một số ứng dụng của bài toàn này là tìm những mẫu thức dữ liệu quan trọng hay những thay đổi bất thường trong dữ liệu ban đầu.

3. CÁC PHƯƠNG PHÁP THU GIẢM SỐ CHIỀU DỰA VÀO ĐẶC TRƯNG

Dữ liệu chuỗi thời gian thường cực kỳ lớn. Tìm kiếm trực tiếp trên những dữ liệu này sẽ rất phức tạp và không hữu hiệu. Để khắc phục vấn đề này, ta nên áp dụng một số phương pháp biến đổi để thu giảm độ lớn của dữ liệu. Những phương pháp biến đổi này thường được gọi là những *kỹ thuật thu giảm số chiều* (dimensionality reduction). Phương pháp tổng quát để thu giảm số chiều có thể tóm tắt như sau:

1. Thiết lập một độ đo tương tự d

2. Thiết kế một kỹ thuật thu giảm số chiều để rút trích một đặc trưng có chiều dài k (tức là một đặc trưng gồm k giá trị), với k có thể được xử lý một cách hữu hiệu nhờ một cấu trúc chỉ mục không gian (đa chiều).

3. Cung cấp một độ đo tương tự d_k trên một không gian đặc trưng k chiều và chứng tỏ rằng nó tuân thủ điều kiện sau đây: $d_k(X', Y') \leq d(X, Y)$ (1)

Điều kiện (1) có nghĩa là hàm khoảng cách tính trên không gian đặc trưng (hay không gian thu giảm) của hai chuỗi thời gian đã được biến đổi X', Y' từ hai chuỗi thời gian ban đầu X, Y phải chặn dưới khoảng cách thật giữa chúng trên không gian nguyên thủy. Điều kiện (1) thường được gọi là *điều kiện chặn dưới*.

Có hai nhóm phương pháp chính để thu giảm số chiều là phương pháp biến đổi sang miền tần số và phương pháp xấp xỉ tuyến tính từng đoạn.

3.1. Các phương pháp biến đổi sang miền tần số

Phương pháp biến đổi fourier rời rạc (discrete Fourier tranform – DFT):

Phương pháp biến đổi rời rạc *Fourier* do *R. Agrawal* và cộng sự đề nghị ([1],[2],[7]). Trong phương pháp biến đổi *Fourier* thì đường dữ liệu ban đầu được biểu diễn bởi các đường căn bản. Nhưng đường căn bản trong trường hợp này là đường *sin* và *cosin*.

$$C(t)=\sum_{k=1}^{n}(A_k\cos(2\pi w_k t)+B_k\sin(2\pi w_k t))$$

Ngoài khả năng nén dữ liệu, với cách tính khoảng cách dựa trên khoảng cách Euclid thì phương pháp *Fourier* cho phép so sánh gián tiếp 2 chuỗi *X, Y* thông qua khoảng cách của 2 chuỗi X_f, Y_f đã được biến đổi. Sở dĩ phép biến đổi *Fourier* rời rạc có tính chất trên là vì:

$$D(X, Y) \geq \alpha\, D(X_f, Y_f)$$

(trong đó α là hằng số)

Vì vậy, phương pháp biến đổi *Fourier* đã được sử dụng trong nhiều ứng dụng và một số phương pháp lập chỉ mục như *F-index*, *ST-index* … đã được đề nghị ([1],[2],[7]). Ưu điểm của phương pháp DFT là thích hợp với các loại đường biểu diễn dữ liệu khác nhau và giải thuật để biến đổi dữ liệu chỉ có độ phức tạp *O(nlgn)*. Tuy nhiên, nhược điểm lớn nhất rất khó giải quyết nếu các chuỗi thời gian có chiều dài khác nhau.

Phương pháp biến đổi wavelet rời rạc (discrete wavelet transform - DWT)

Phương pháp thu giảm số chiều bằng biến đổi rời rạc *Wavelet* rời rạc do *K. Chan* và *W. Fu* đề nghị năm 1999 ([5]). Phương pháp *DWT* cũng giống phương pháp *DFT*, tuy nhiên đường cơ bản của nó không phải là đường lượng giác *sin* hay *cosin* mà là đường *Haar*. Đường *Haar* được định nghĩa theo các công thức ψ_j^i như sau:

$$\psi_f^i = \psi(2^j x - i)$$

$i = 0,\ldots 2^j - 1$

với $\psi_j^i(t)$ =

$$\begin{cases} 1 \text{ với } 0< t <0.5 \\ -1 \text{ với } 0.5< t <1 \\ 0 \text{ các trường hợp khác} \end{cases}$$

Ngoài sử dụng đường *Haar*, phương pháp *Wavelet* có thể sử dụng các đường cơ bản khác như đường *Daubechies, Coiflet, Symmlet*… Tuy nhiên, *Haar Wavalet* đã được sử dụng rất nhiều trong khai phá dữ liệu chuỗi thời gian và lập chỉ mục ([14]).

Phương pháp biến đổi *Wavelet* rời rạc rất hiệu quả bởi vì nó mã hóa đơn giản và nhanh. Độ phức tạp của việc mã hóa này là tuyến tính. Một ưu điểm của phương pháp Wavelet là nó hỗ trợ nhiều mức phân giải.

Khi áp dụng kỹ thuật thu giảm số chiều bằng DFT hay DWT, thủ tục so trùng chuỗi con được tiến hành bằng các bước sau:

1. Với mỗi vị trí trên chuỗi thời gian ban đầu, dùng một *cửa sổ trượt* (sliding window) có kích thước *w*, và tạo ra một đặc trưng chiều dài *k* (một điểm trong không gian thu giảm k chiều). Mỗi điểm này sẽ gần với điểm đi trước vì nội dung của cửa số trượt thay đổi chậm. Những điểm trong dữ liệu chuỗi thời gian ban đầu sẽ được biến đổi thành một *vết* (trail) trong không gian đặc trưng.

2. Phân đoạn *vết* thành những *vết con* (subtrail). Lưu mỗi vết con trong *hình chữ nhật bao nhỏ nhất* (minimal bouding rectangle-MBR) đa chiều.

3. Lưu các MBR trong một cấu trúc chỉ mục không gian.

Để truy tìm những chuỗi con tương tự với chuỗi con truy vấn *Q* có chiều dài *w,* ta chỉ cần rà soát tất cả các MBR có giao với hình cầu đa chiều (hypersphere) có bán kính *r* (*r*: là ngưỡng sai số cho phép) bao quanh điểm đặc trưng *Q'* mà thể hiện chuỗi con truy vấn *Q*.

Các cấu trúc chỉ mục không gian thông dụng để hỗ trợ tìm kiếm tương tự trên dữ liệu chuỗi thời gian có thể kể như cây R* ([3]), cây M ([6]), v.v.

3.2. Các phương pháp xấp xỉ tuyến tính từng đoạn

Phương pháp xấp xỉ tuyến tính từng đoạn

Phương pháp *xấp xỉ tuyến tính từng đoạn* (piecewise linear approximation- PLA**)** do E. Keogh và cộng sự đề nghị ([8],[9]). Trong phương pháp này ta sẽ biểu diễn dữ liệu ban đầu bằng chuỗi các đoạn thẳng tuyến tính. Mỗi đoạn thẳng tuyến tính nối cặp điểm ở hai đầu đoạn thẳng xấp xỉ tốt nhất (best-fit) những điểm có trong phân đoạn chuỗi thời gian đó. Các đoạn thẳng này có thể rời nhau hoặc liên tục. Cách biểu diễn này rất trực quan và nó phù hợp để nén tất cả các loại dữ liệu chuỗi thời gian.

Phương pháp xấp xỉ gộp từng đoạn (PAA)

Phương pháp *xấp xỉ gộp từng đoạn* (piecewise aggregate approximation) do E. Keogh và cộng sự đề nghị năm 2001([10]). Phương pháp này rất đơn giản, ta tuần tự xấp xỉ *k* giá trị liền kề nhau thành cùng một giá trị bằng trung bình cộng của *k* điểm đó. Qúa trình cứ tiếp tục như vây từ trái sang phải.

Với phương pháp này, thời gian tính toán rất nhanh và cách biểu diễn của nó hỗ trợ nhiều độ đo khoảng cách.

Phương pháp xấp xỉ hằng số từng đoạn thích nghi

Phương pháp *xấp xỉ hằng số từng đoạn thích nghi* (adaptive piecewise constant approximation *–APCA*) do *E. Keogh* và cộng sự đề nghị năm 2001 ([11]). Phương pháp *APCA* giống như phương pháp *PAA* là xấp xỉ dữ liệu ban đầu thành những đoạn thẳng nằm ngang. Tuy nhiên, nó khác với *PAA* là các đoạn này ở PAA có kích thước bằng nhau, còn ở *APCA* thì kích thước của các đoạn là khác nhau tùy theo dữ liệu. Những vùng nào trên chuỗi thời gian có biến động nhấp nhô nhiều thì được phân thành những đoạn ngắn, còn những vùng nào ít biến động thì được phân thành những đoạn dài hơn.

4. CÁC PHƯƠNG PHÁP RỜI RẠC HÓA

Trong *phương pháp rời rạc hóa* (discretization) thì từ dữ liệu ban đầu, ta sẽ chia thành những đoạn dữ liệu nhỏ hơn, rồi sau đó, tương ứng với mỗi đoạn nhỏ này ta sẽ mã hóa chúng bởi những đặc trưng của đoạn và tập hợp những đặc trưng của những đoạn nhỏ này sẽ làm thành một tràng ký hiệu biểu diễn cho dữ liệu ban đầu. Có một số phương pháp rời rạc hóa đã được đề xuất, tiêu biểu nhất là SAX và iSAX. Lợi ích quan trọng của việc rời rạc hóa dữ liệu chuỗi thời gian là điều này cho phép sử dụng những cấu trúc dữ liệu và giải thuật vốn có trong lãnh vực xử lý chuỗi ký tự như cây hậu tố, kỹ thuật băm, mô hình xích Markov,v.v…

4.1 Phương pháp xấp xỉ gộp ký hiệu hóa - SAX

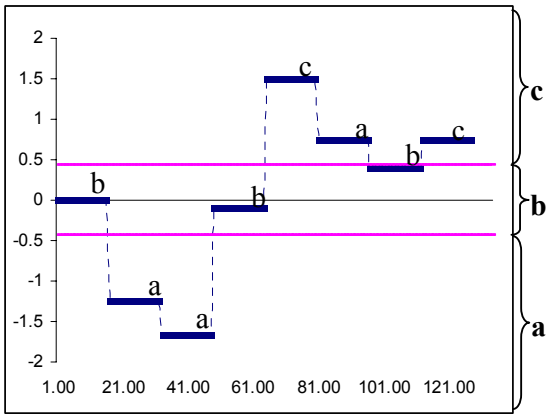
Lin, Keogh và các cộng sự ([13]) đã đề xuất một phương pháp rời rạc hóa có tên là *xấp xỉ gộp ký hiệu hóa* (symbolic aggregate approximation – SAX) mà dựa trên phương pháp thu giảm số chiều PAA và giả sử dữ liệu thu giảm số chiều đã được chuẩn hóa. SAX là quá trình ánh xạ biểu diễn PAA của chuỗi thời gian thành một chuỗi ký tự rời rạc. Gọi a là kích thước của bộ ký hiệu mà được dùng để rời rạc hóa chuỗi thời gian. Để ký hiệu hóa chuỗi thời gian chúng ta phải tìm thấy các trị (điểm ngắt) sau đây:

$$\beta_1, \beta_2, ..., \beta_{a-1} \text{ với } \beta_1 < \beta_2 < ... < \beta_{a-1}$$

Sử dụng những trị này, chuỗi thời gian $\overline{T} = \overline{t_1}, ..., \overline{t_w}$ sẽ được rời rạc hóa thành tràng ký hiệu $C = c_1 c_2 .... c_w$. Mỗi đoạn $\overline{t_i}$ sẽ được mã hóa thành ký hiệu c_i dùng công thức sau:

$$c_i = \begin{cases} c_1 & \overline{t_i} \le \beta_1 \\ c_a & \overline{t_i} > \beta_{a-1} \\ c_k & \beta_{k-1} < \overline{t_i} \le \beta_k \end{cases} \quad (2)$$

Hình 2 minh họa phương pháp mã hóa SAX.



Hình 2.Một chuỗi thời gian được biến đổi PAA rồi mã hóa thành các ký hiệu SAX. Chuỗi thời gian được mã hóa thành baabccbc.

Chúng ta chú ý rằng dữ liệu chuỗi thời gian thường có phân bố xác xuất Gauss. Để có một xác xuất bằng nhau (1/a) cho mỗi ký hiệu, ta phải chọn trị *điểm ngắt* β_i dựa trên bảng xác xuất của phân bố Gauss.

Sau giai đoạn mã hóa SAX, thì bài toán so trùng chuỗi thời gian trở thành bài toán so trùng chuỗi ký tự. Để thực hiện việc so trùng này nhanh chóng thì cấu trúc *cây hậu tố* (suffix tree) hay *tập tin VA* (vector approximation file) có thể được sử dụng làm cấu trúc chỉ mục.

4.2 Phương pháp iSAX

Phương pháp rời rạc hóa iSAX (indexable symbolic aggregate approximation) được Shieh và Keogh đưa ra năm 2008 ([15]). Phương pháp này là sự mở rộng của phương pháp SAX mà thay vì mã hóa mỗi phân đoạn chuỗi thời gian thành một ký hiệu như là chữ hay số nguyên, iSAX mã hóa nó thành số nhị phân, thí dụ “00”, “01”,”10”,”11”.

Với cách này, iSAX có thể mã hóa chuỗi thời gian thành chuỗi bit và nhờ đó tiết kiệm

được rất nhiều chỗ bộ nhớ lưu trữ. Ngoài ra, iSAX còn hỗ trợ khả năng biểu diễn *đa mức phân giải* (multi-resolution) dựa trên khả năng đa mức phân giải của các mức rời rạc hóa. Bằng việc kết hợp với một cấu trúc chỉ mục *cây phân cấp* (hierarchical tree) với phương pháp iSAX, ta có thể truy vấn nhanh trên cơ sở dữ liệu chuỗi thời gian với kích thước lớn lên đến hàng terabyte.

5. KẾT LUẬN

Bài viết này điểm qua những nghiên cứu gần đây của lãnh vực tìm kiếm tương tự trên dữ liệu chuỗi thời gian. Những nghiên cứu này hội tụ vào phương thức tìm kiếm tương tự dựa trên cách tiếp cận rút trích đặc trưng để thu giảm số chiều. Có 2 nhóm phương pháp rút trích đặc trưng chính được trình bày: các phép biến đổi sang miền tần số, và phương pháp xấp xỉ tuyến tính từng đoạn. Cuối cùng, các phương pháp rời rạc hóa nhằm mã hóa chuỗi thời gian thành chuỗi ký hiệu cũng được trình bày.

Mặc dù lãnh vực tìm kiếm tương tự dữ liệu chuỗi thời gian đã thu hút được sự quan tâm của giới nghiên cứu và là một lãnh vực tương đối trưởng thành, nhưng vẫn còn nhiều vấn đề cần phải được nghiên cứu thêm. Có bốn lãnh vực như vậy có thể kể: (1) nghiên cứu so sánh bằng thực nghiệm hiệu quả của các phương pháp tìm kiếm tương tự đã đề xuất, (2) nghiên cứu so sánh bằng thực nghiệm hiệu quả của các độ đo tương tự khác nhau đã đề xuất (3) đề xuất các cấu trúc chỉ mục hữu hiệu hơn cho dữ liệu chuỗi thời gian và (4) ứng dụng của các phương pháp này trong các lãnh vực khai phá dữ liệu chuỗi thời gian (time series data mining).

AN OVERVIEW OF SIMILARITY SEARCH IN TIME SERIES DATA

Duong Tuan Anh

University of Technology, VNU-HCM

ABSTRACT**:** *Time series data occur in many real life applications, ranging from science and engineering to business. In many of these applications, searching through large time series database based on query sequence is often desirable. Such similarity-based retrieval is also the basic subroutine in several advanced time series data mining tasks such as clustering, classification, finding motifs, detecting anomaly patterns, rule discovery and visualization. Although several different approaches have been developed, most are based on the common premise of dimensionality reduction and spatial access methods. This survey gives an overview of recent research and shows how the methods fit into a general framework of feature extraction.*

Keywords*: time series, dimensionality reduction, discretization, feature extraction.*

TÀI LIỆU THAM KHẢO

[1]. Agrawal, R., Faloutsos, C. and Swami, A.N., 1993, Efficient Similarity Search in Sequence Databases, *Proc. 4th Int. Conf. on Foundations of Data Organization and Algorithms (FODO),* pp. 69-84.

[2]. Agrawal, R., Lin, K., Sawhney, H.S., and Shim, K., 1995, Fast similarity search in the presence of noise, scaling, and translation in time-series databases. *In Proceedings of the 21st International Conference on Very Large Databases,* VLDB95, Zurich, Switzerland, pp. 490-501.

[3]. Beckmann N., et al., 1990, The R*-tree: an efficient and robust access method for points and rectangles, *In Proceedings of the ACM SIGMOD International Conference on Management of Data*, SIGMOD90, Atlantic City, NewYork, USA, pp. 322- 331.

[4]. Berndt, D. and Clifford J., 1994, Using dynamic time warping to find patterns in time series. *In Proceedings of AAAI Workshop on Knowledge Discovery in Databases*, KDD-94, Seattle, Washington, USA, pp. 359-370.

[5]. Chan, K., Fu, A. W., 1999, Efficient time series matching by wavelets. *In Proceedings of the 15th IEEE International Conference on Data Engineering*, Sydney, Australia, pp. 126-133.

[6]. Ciaccia P., et al., 1997, M-tree: An Efficient Access Method for Similarity Search in Metric Spaces, *In Proceedings of the 23rd VLDB International Conference*, Athens, Greece, pp. 426-435.

[7]. Faloutsos, C., Ranganathan, M., Manolopoulos, Y., Fast Subsequence Matching in Time-Series Databases, *Proceedings of the 14th ACM SIGMOD International Conference on Management of Data (SIGMOD 1994),* May 24-27, 1994, pp. 419-429.

[8]. Keogh E. and Pazzani M., 1998, An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback, *In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, New York, NY, Aug 27-31. pp 239-241.

[9]. Keogh E., et al., 2001, An online algorithm for segmenting time series. *In Proceedings of the IEEE International Conference on Data Mining*, California, USA, pp. 289-296.

[10]. Keogh, E., Chakrabarti, K., Pazzani,M. and Mehrotra, S., 2001, Dimensionality reduction for fast similarity search in large time series databases, *Journal of Knowledge and Information Systems*, Vol. 3, No. 3, 2000, pp. 263-286.

[11]. Keogh, E., Chakrabarti, K., Pazzani,M. and Mehrotra, S, 2001, Locally adaptive dimensionality reduction for

indexing large time series databases, *Proceedings of the 2001 ACM SIGMOD Conference on Management of Data,* May 21-24, 2001, pp. 151-162.

[12]. Keogh E., 2006, A Tutorial on Indexing and Mining Time Series Data, In *Proceedings of the 32th International Conference on Very Large Databases, VLDB2006*, Seoul, Korea.

[13]. Lin J., Keogh, E., Lonardi, S., and Chiu, B., 2003, A Symbolic Representation of Time Series, with Implications for Streaming Algorithms, *In Proceedings of 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discover*, DMKD 2003, California, USA, pp. 2-11.

[14]. Popivanov I., Miller R.J., 2002, Efficient Similarity Queries Over Time Series Data Using Wavelets, *In Proceedings of the 18th International Conference on Data Engineering*, San Jose, California, USA, pp. 212 - 221.

[15]. Shieh, J. and Keogh, E., 2008, iSAX: Indexing and Mining Terabyte sized Time Series, *Proc. of SIGKDD 2008*.