PHÁT HIỆN MOTIF TRÊN CHUỖI THỜI GIAN BẰNG CẤU TRÚC CHỈ MỤC ĐA CHIỀU

DISCOVERING MOTIFS IN TIME SERIES WITH MULTI-DIMENSIONAL INDEX STRUCTURE

Nguyễn Thành Sơn
Trường Đại học Sư phạm Kỹ thuật TP.HCM

TÓM TẮT

Motif trong cơ sở dữ liệu chuỗi thời gian là các chuỗi lặp lại nhiều lần trong cơ sở dữ liệu chuỗi thời gian hoặc các chuỗi con lặp lại trong một chuỗi thời gian dài hơn. Phát hiện motif trên chuỗi thời gian là một công việc quan trọng trong khai phá dữ liệu chuỗi thời gian. Trong bài báo này, chúng tôi đề xuất một phương pháp phát hiện motif trên chuỗi thời gian dựa vào một cấu trúc chỉ mục đa chiều sử dụng vùng bao hình chữ nhật nhỏ nhất. Phương pháp do chúng tôi đề xuất hiệu quả về mặt thời gian xử lý lẫn không gian lưu trữ vì chỉ cần lưu các vùng bao nhỏ nhất của các chuỗi thời gian trong bộ nhớ chính và chỉ cần quét qua một lần toàn bộ cơ sở dữ liệu chuỗi thời gian cùng với môt vài lần đọc dữ liệu gốc từ đĩa để thẩm định lại kết quả. Chúng tôi minh họa tính hiệu quả của phương pháp đề xuất bằng thực nghiệm trên các tập dữ liệu thực thuộc các lĩnh vực khác nhau. Kết quả thực nghiệm cho thấy phương pháp đề xuất có thể phát hiện motif một cách hiệu quả hơn những phương pháp thông dụng, phương pháp chiếu ngẫu nhiên.

***Từ khóa**: Chuỗi thời gian, chỉ mục đa chiều, motif.*

ABSTRACT

Time series motifs are frequently occurring but unknown sequences in time series database or subsequences of a longer time series. Discovering time series motifs is a crucial task in time series data mining. In this paper, we examine a search method for discovering approximate motif in time series with the support of a multidimensional index structure based on Minimum Bounding Rectangles (MBR). Our method is time and space efficient because it only saves MBRs of data in the memory and needs a single scan over the entire time series database and a few times to read the original disk data in order to confirm the results. We demonstrate the effectiveness of our approach by experimenting on real datasets from different areas. The experimental results showed that our proposed method can effectively discover time series motifs as compared to the popular method, random projection.

***Key words**: Time series, Multi-dimensional index, motif*

I. GIỚI THIỆU

Tìm kiếm motif trên dữ liệu chuỗi thời gian (time series data) là một công việc quan trọng trong nhiều lĩnh vực nghiên cứu khác nhau như gom cụm dữ liệu chuỗi thời gian, phân lớp dữ liệu chuỗi thời gian, khám phá luật kết hợp trong dữ liệu chuỗi thời gian [8], phân tích cấu trúc video [2][16], phát hiện bất thường trong dữ liệu chuỗi thời gian, dự báo ([5]).

Tùy thuộc vào dữ liệu có được thu giảm số chiều hay không, các phương pháp tìm kiếm motif trên dữ liệu chuỗi thời gian được phân thành hai nhóm: các phương pháp tìm kiếm chính xác và các phương pháp tìm kiếm xấp xỉ.

- Các phương pháp tìm kiếm chính xác

motif phân tích trực tiếp trên dữ liệu gốc.

- Các phương pháp tìm kiếm xấp xỉ phân tích dữ liệu trong không gian thu giảm. Các phương pháp này thường dùng các kỹ thuật xử lý trên chuỗi ký tự mà không phân tích trực tiếp trên dữ liệu số.

Độ phức tạp của các thuật toán tìm kiếm motif xấp xỉ thường là $O(n)$ hoặc $O(nlogn)$ với một số lớn các hệ số phải xác định trước. Các phương pháp tìm kiếm motif xấp xỉ thường dựa trên các kỹ thuật xử lý chuỗi ký tự vì vậy người ta thường nghiên cứu các phương pháp biểu diễn ký tự khác nhau để chuyển đổi dữ liệu chuỗi thời gian thành dạng chuỗi ký tự. Tuy nhiên các kỹ thuật xử lý chuỗi ký tự không thể trực tiếp phân tích trên dữ liệu chuỗi thời gian dạng số.

Mặc dù có những nghiên cứu gần đây về phương pháp tìm kiếm motif chính xác, chúng tôi tin rằng cách tiếp cận tìm kiếm motif xấp xỉ vẫn tiếp tục là lựa chọn tốt nhất trong nhiều ứng dụng ở các lĩnh vực khác nhau do tính hiệu quả về mặt thời gian và/hoặc không gian của nó. Hơn nữa cách tiếp cận tìm kiếm gần đúng motif có thể phân tích trực tiếp trên dữ liệu chuỗi thời gian dạng số vẫn còn là một thách thức khó khăn. Điều đó thúc đẩy chúng tôi nghiên cứu một phương pháp mạnh và hiệu quả theo hướng tiếp cận này.

Trong bài báo này, chúng tôi giới thiệu phương pháp tìm kiếm motif trên dữ liệu chuỗi thời gian bằng R*-tree dựa trên vùng bao MBR. Phương pháp này hiệu quả vì chỉ cần một lần đọc qua cơ sở dữ liệu (CSDL) và một vài lần truy cập chuỗi gốc để thẩm định kết quả và trong bộ nhớ chỉ cần lưu các MBR của chuỗi thời gian. Trong thực nghiệm chúng tôi so sánh phương pháp này với phương pháp chiếu ngẫu nhiên vì đây là phương pháp thông dụng và thường là cơ sở cho nhiều cách tiếp cận khác nhau về tìm kiếm motif [10],[15],[17]. Hơn nữa, thời gian thao tác của thuật toán này có thể được xem như chặn dưới của thời gian thao tác của các cách tiếp cận dựa trên phương pháp chiếu này. Để chứng tỏ tính hiệu quả của phương pháp này chúng tôi tiến hành thực nghiệm trên các tập dữ liệu thực thuộc lĩnh vực khác nhau. Kết quả thực nghiệm cho thấy cách tiếp cận này hiệu quả hơn so với phương pháp chiếu ngẫu nhiên.

Phần còn lại của bài báo được tổ chức như sau. Phần 2 giới thiệu về các nghiên cứu trước đây và các khái niệm liên quan. Phương pháp chúng tôi đề xuất được trình bày ở phần 3. Kết quả thực nghiệm được báo cáo trong phần 4. Phần 5 là kết luận và hướng phát triển.

II. CÁC NGHIÊN CỨU TRƯỚC ĐÂY VÀ KIẾN THỨC LIÊN QUAN

1. Các nghiên cứu trước đây

Trong phần này chúng tôi trình bày tóm tắt một số phương pháp tìm kiếm motif trên chuỗi thời gian đã được giới thiệu.

- **Các kỹ thuật tìm kiếm motif xấp xỉ**

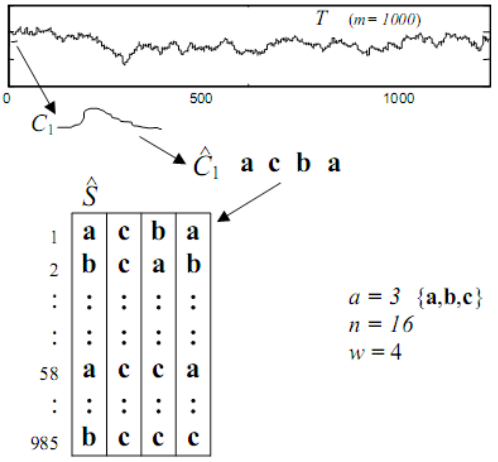
Cách tiếp cận chung của các phương pháp tìm kiếm motif xấp xỉ là dùng các kỹ thuật xử lý chuỗi để phát hiện motif. Với cách tiếp cận này, đầu tiên dữ liệu chuỗi thời gian gốc được biến đổi thành chuỗi ký tự sau đó dùng các thuật toán khai phá chuỗi ký tự để tìm motif.

Nhiều thuật toán tìm kiếm motif trong dữ liệu chuỗi thời gian đã được giới thiệu từ khi bài toán được xác định vào năm 2002 [8]. Trong [8] Lin và các cộng sự định nghĩa bài toán phát hiện motif trên chuỗi thời gian dựa vào một ngưỡng R và một chiều dài motif m do người dùng xác định. Theo đó hai chuỗi con c_i bắt đầu ở vị trí i và c_j bắt đầu ở vị trí j có chiều dài m trong một chuỗi thời gian có chiều dài n ($m << n$) tạo thành một cặp tương tự không tầm thường (non-trivial matching) nếu độ đo tương tự giữa chúng DISTANCE(C_i, C_j) < R và tồn tại một chuỗi con C_k bắt đầu tại vị trí k mà DISTANCE(C_i, C_k) > R và $i < k < j$ hoặc $j < k < i$. Khái niệm tương tự này sau đó được mở rộng thành bài toán phát hiện những motif bậc k hàng đầu trên chuỗi thời gian, trong đó motif bậc nhất

trên chuỗi thời gian là chuỗi con c_1 có số chuỗi con tương tự không tầm thường nhiều nhất. Motif bậc k là chuỗi con c_k có số chuỗi con tương tự không tầm thường nhiều thứ k và thỏa DISTANCE(C_i, C_k) > 2R, với mọi $1 \leq i < k$.

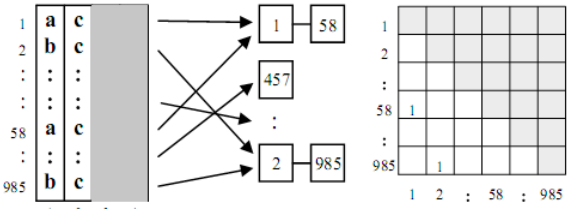
Trong [3] Chiu và các cộng sự đề xuất giải thuật chiếu ngẫu nhiên để phát hiện motif trên chuỗi thời gian theo cách tiếp cận xấp xỉ. Giải thuật này dựa trên kỹ thuật *băm bảo toàn tính lân cận* (locality preserving hashing). Kỹ thuật này sử dụng phương pháp rời rạc hóa SAX để biểu diễn các chuỗi con trong chuỗi thời gian ban đầu và một ma trận đụng độ có số dòng và cột là số chuỗi con được trích từ chuỗi thời gian ban đầu. Mỗi vòng lặp, thuật toán sẽ lựa chọn ngẫu nhiên một số vị trí trong biểu diễn SAX để làm mặt nạ và duyệt qua danh sách biểu diễn SAX. Nếu hai chuỗi biểu diễn SAX tương ứng với hai chuỗi con i và j giống nhau thì ô (i, j) trong ma trận đụng độ sẽ được tăng lên một.

Sau khi tiến trình trên được lặp lại một số lần thích hợp, các ô có giá trị lớn trong ma trận đụng độ sẽ được chọn làm các ứng viên motif. Cuối cùng dữ liệu gốc tương ứng với các ứng viên motif sẽ được kiểm tra để thẩm định kết quả. Hình 1 là một ví dụ minh họa một chuỗi thời gian có chiều dài 1000 điểm và biểu diễn SAX của các chuỗi con có chiều dài n = 16, chiều dài bộ ký tự SAX là a = 3, số chiều w của chuỗi con sau khi thu giảm theo PAA là 4.



Hình 1. Ví dụ minh họa một chuỗi thời gian T và biểu diễn SAX của các chuỗi con của T.

Hình 2 là một ví dụ minh họa thực hiện lần lặp thứ nhất của phương pháp chiếu ngẫu nhiên trên các chuỗi SAX được minh họa ở hình 1. Trong ví dụ này hai cột một và hai được chọn ngẫu nhiên (hình phía trái), được dùng để tạo ma trận đụng độ (hình phía phải). Trong ví dụ ta có hai cặp chuỗi con giống nhau là (1, 58) và (2, 985), do đó hai ô tương ứng với hai cặp chuỗi con này trong ma trận đụng độ được tăng lên một. Độ phức tạp của thuật toán này là tuyến tính theo độ dài của từ SAX, số chuỗi con, số lần lặp và số lần đụng độ [10].



Hình 2. Ví dụ minh họa lần lặp thứ nhất của giải thuật chiếu ngẫu nhiên.

Thuật toán này đã được sử dụng rộng rãi để phát hiện motif trên chuỗi thời gian từ khi nó được giới thiệu và có thể được dùng để phát hiện tất cả motif với xác xuất cao sau một số lần lặp thích hợp ngay cả trong trường hợp có nhiễu. Tuy nhiên, nó vẫn có những nhược điểm sau: (1) để thực hiện thuật toán, nhiều tham số nhập cần phải được xác định trước bởi người sử dụng, (2) độ phức tạp của thuật toán này sẽ trở thành bậc hai nếu sự phân bố của phép chiếu không đủ rộng, nghĩa là có một số lớn các chuỗi con có cùng kết quả chiếu [10].

Trong [15], các tác giả sử dụng SAX cho trường hợp dữ liệu chuỗi thời gian nhiều biến bằng cách dùng phương pháp PCA (Principle Component Analysis) để biến đổi dữ liệu chuỗi thời gian nhiều biến thành một biến. Sau đó sử dụng phép chiếu ngẫu nhiên để tìm kiếm motif.

Một kỹ thuật tìm kiếm motif xấp xỉ khác được giới thiệu trong [6]. Đầu tiên, thuật toán này biến đổi các chuỗi con trong dữ liệu chuỗi thời gian thuộc lĩnh vực Proteins theo dạng

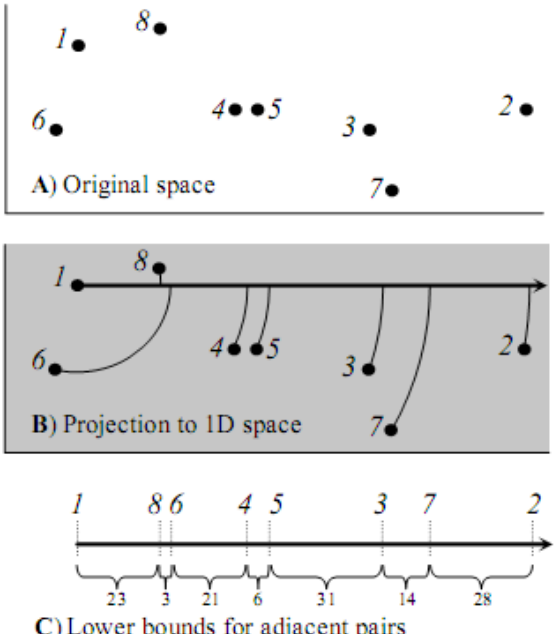
biểu diễn SAX. Sau đó thuật toán tìm kiếm motif bằng cách gom cụm các chuỗi con và mở rộng mỗi chiều dài motif thu được cho tới khi độ đo tương tự nhỏ hơn hoặc bằng một ngưỡng do người dùng định nghĩa.

Năm 2010, Castro và Azevedo đã giới thiệu một phương pháp phát hiện motif xấp xỉ, gọi là MrMotif (Multiresolution Motif) [4]. Thuật toán này dựa vào phương pháp rời rạc hóa iSAX [14] và thuật toán tiết kiệm không gian [12]. Ý tưởng chính của thuật toán này như sau: bắt đầu từ biểu diễn iSAX ở mức phân giải thấp của các chuỗi thời gian, sau đó mở rộng dần lên các mức phân giải cao hơn. Ở mỗi mức phân giải, thuật toán tiến hành gom cụm các chuỗi theo biểu diễn iSAX giống nhau. Số chuỗi trong mỗi cụm sẽ giảm đi khi mỗi cụm ở mức phân giải thấp được chia thành nhiều cụm ở mức phân giải cao hơn. Tại mức phân giải cao nhất, các chuỗi trong một cụm sẽ tương tự nhau nhất. Thuật toán này cho phép phát hiện motif ở nhiều mức phân giải khác nhau và có thể áp dụng cho dữ liệu dạng luồng. Tuy nhiên, phương pháp này vẫn phải trải qua giai đoạn rời rạc hóa mà chưa thể làm việc thật tiện lợi trên dữ liệu chuỗi thời gian dạng số và ở mỗi mức phân giải khác nhau sẽ cho ra kết quả phát hiện motif khác nhau vì độ chính xác của biểu diễn ở mỗi mức phân giải là khác nhau.

- **Các kỹ thuật tìm kiếm motif chính xác**

Các kỹ thuật tìm kiếm motif chính xác bỏ qua giai đoạn ký tự hóa dữ liệu chuỗi thời gian bằng cách phân tích trực tiếp trên dữ liệu gốc. Năm 2002, Oates đề xuất một thuật toán tìm kiếm những mẫu lặp lại trong dữ liệu chuỗi thời gian nhiều biến được gọi là PERUSE [13]. Thuật toán này phân tích trực tiếp trên dữ liệu gốc bằng cách dùng cửa sổ trượt quét qua toàn bộ dữ liệu và lập trình động để tìm motifs. Nó có thể xử lý dữ liệu được lấy mẫu ở các tần số khác nhau và các mẫu lặp có chiều dài bất kỳ.

Trong [11] Abdullah Mueen và các cộng sự đề xuất một thuật toán tìm kiếm chính xác motif, gọi là thuật toán MK. Cách tiếp cận này sử dụng các điểm tham chiếu được chọn ngẫu nhiên và ý tưởng từ bỏ sớm việc tính toán khoảng cách Euclid khi tổng tích lũy khoảng cách hiện hành lớn hơn khoảng cách của ứng viên motif tốt nhất tại thời điểm đang xét. Quá trình phát hiện motif của thuật toán này dựa vào thông tin heuristic được xác định bởi thứ tự của khoảng cách giữa đối tượng đang xét với các điểm tham chiếu ngẫu nhiên. Hình 3 là một ví dụ minh họa ý tưởng sử dụng điểm tham chiếu. Hình 3a là một tập các đối tượng chuỗi thời gian hai chiều. Giả sử đối tượng số 1 được chọn làm điểm tham chiếu. Các đối tượng khác được sắp thứ tự theo khoảng cách của chúng tới điểm tham chiếu số 1 trong không gian một chiều (Hình 3B và 3C). Các khoảng cách này là các chặn dưới của các khoảng cách thực tương ứng của chúng trong không gian gốc.



Hình 3. Một ví dụ minh họa ý tưởng sử dụng điểm tham chiếu.

Thứ tự của các đối tượng trong không gian một chiều cung cấp thông tin heuristic hữu ích hướng dẫn việc phát hiện motif. Nếu hai đối tượng gần nhau trong không gian gốc, chúng cũng phải gần nhau theo thứ tự trong không gian một chiều. Nhưng ngược lại thì không đúng, nghĩa là hai đối tượng có thể

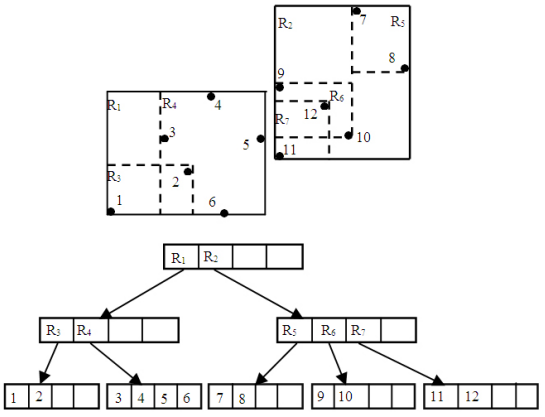
gần nhau theo thứ tự trong không gian một chiều nhưng lại rất xa trong không gian gốc. Thuật toán dùng một biến *BestSoFar* để lưu khoảng cách của một cặp ứng viên motif tốt nhất tính đến thời điểm đang xét. Khi thuật toán duyệt qua thứ tự trong không gian một chiều, nếu có một cặp chuỗi có khoảng cách nhỏ hơn *BestSoFar* hiện hành thì *BestSoFar* được cập nhật lại theo giá trị mới. Giá trị *BestSoFar* này cùng với biểu diễn thứ tự các đối tượng trong không gian một chiều theo khoảng cách của chúng tới điểm tham chiếu sẽ giúp loại bỏ phần lớn không gian tìm kiếm.

Thuật toán MK là một sự cải tiến của thuật toán brute-force bằng cách sử dụng hai kỹ thuật trên để giảm thiểu thời gian thực hiện của thuật toán. Mueen và các cộng sự đã cho thấy rằng thuật toán này có thể thực hiện nhanh hơn gấp vài ngàn lần thuật toán brute-force khi tìm trong cơ sở dữ liệu lớn, dù trong trường hợp xấu nhất độ phức tạp của thuật toán là bậc hai. Nhược điểm của MK là: (1) do sử dụng độ đo Euclid trực tiếp trên dữ liệu thô nên giải thuật MK dễ bị tác động bởi nhiễu, (2) do MK dựa vào sự vét cạn của giải thuật brute-force kết hợp với một số kỹ thuật tăng tốc, tính hữu hiệu của MK tuy có được cải thiện nhưng vẫn chưa cao như mong muốn.

2. Cấu trúc chỉ mục đa chiều

Cấu trúc chỉ mục đa chiều thông dụng cho chuỗi thời gian là R-tree và các biến thể của nó ([7], [1]). Trong một cấu trúc chỉ mục R-tree, mỗi nút trong cây chứa từ *m* đến *M* phần tử trừ khi nút đó là nút gốc (nút gốc có thể có ít nhất 2 phần tử). Chặn dưới *m* được sử dụng nhằm tránh sự suy biến của cây. Khi số phần tử trong một nút nhỏ hơn *m*, nút đó sẽ bị xóa và các phần tử của nút sẽ được cấp phát lại cho các nút kế cận. Chặn trên *M* nhằm mục đích đảm bảo mỗi một nút có thể lưu trữ được một trang dữ liệu đĩa (disk page). Mỗi phần tử trong một nút không phải lá chứa *một vùng bao chữ nhật nhỏ nhất* (Minimum Bounding Rectangle – MBR) và một con trỏ đến nút con của nó. Một MBR tại phần tử trong một nút là một vùng nhỏ nhất bao các MBR của các nút con của nó. Mỗi phần tử trong nút lá chứa một MBR của chuỗi thời gian và một con trỏ đến đối tượng dữ liệu nguyên thủy được bao bởi MBR.

Điểm yếu của R-tree là các MBR trong các nút trên cùng một mức có thể phủ lấp nhau. Sự phủ lấp (overlap) này có thể làm giảm hiệu quả thực thi của việc tìm kiếm dựa vào chỉ mục. Hình 4 minh họa các MBR và R-tree tương ứng.



Hình 4. Minh họa R-tree.

Tác vụ tìm kiếm trong R-tree tương tự như tác vụ tìm kiếm trong B-tree. Tại mỗi nút nội, các phần tử cùng với nút con của nó sẽ được kiểm tra xem MBR của phần tử đó có giao với vùng bao MBR của chuỗi truy vấn không.

Để chèn một chuỗi mới vào R-tree, giải thuật sẽ chèn vùng bao MBR của chuỗi và con trỏ tới nó vào cây. Giải thuật sẽ duyệt cây dọc theo một lối đi từ nút gốc đến nút lá. Tại mỗi mức, giải thuật lựa chọn phần tử cần mở rộng vùng bao ít nhất khi chèn MBR của chuỗi mới vào. Khi đến nút lá, nếu nút còn đủ chỗ trống giải thuật sẽ chèn vùng bao MBR của chuỗi và con trỏ đến nó vào nút. Ngược lại, giải thuật sẽ tiến hành tách nút. Tiến trình tách nút có thể được lan truyền ngược từ nút lá lên trên nếu nút cha của nút bị tách không còn chỗ trống.

R*-tree là một biến thể của R-tree, do

Beckmann và các cộng sự đề xuất năm 1990. Các tác giả đã cải tiến tác vụ chèn thêm đối tượng mới vào cây của R-tree bằng cách sử dụng kỹ thuật tách nút dựa trên các tiêu chuẩn tối ưu hóa [1].

3. Các định nghĩa liên quan

Trong phần này chúng tôi trình bày các định nghĩa về motif được dùng trong nghiên cứu của chúng tôi.

- **Định nghĩa 1**. Motif trong CSDL chuỗi thời gian D là một cặp chuỗi thời gian khác nhau $\{T_i, T_j\}$, $i \neq j$, trong D có khoảng cách nhỏ nhất. i.e. $\forall x, y, x \neq y$, $Distance(T_i, T_j) \leq Distance(T_x, T_y)$

Định nghĩa 1 có thể được tổng quát hóa bằng định nghĩa k motifs đầu tiên và cụm motif.

- **Định nghĩa 2**. k motifs đầu tiên là một tập có thứ tự $S = \{M_1, M_2, \ldots, M_k\}$ gồm k cặp chuỗi thời gian không giao nhau trong CSDL D, i.e $M_i = \{T_{i1}, T_{i2}\}, i_1 \neq i_2, M_1 \cap M_2 \cap \ldots \cap M_k = \varnothing$, trong đó $Distance(M_1) \leq Distance(M_2) \leq \ldots \leq Distance(M_k)$ và $\forall x, y, x \neq y, T_x, T_y$ Î $D, \{T_x, T_y\} \notin S, Distance(M_k) \leq Distance(T_x, T_y)$.

- **Định nghĩa 3**. Cụm motif với ngưỡng □ là một tập S có số chuỗi thời gian lớn nhất thỏa điều kiện: $\forall T_i, T_j \in S$, $Distance(T_i, T_j) \leq 2$□ and $\forall T_x \in D\text{-}S$, $Distance(T_i, T_x) > 2$□ ([11])

Các định nghĩa cho trường hợp motif là các chuỗi con tương tự có chiều dài m trong một chuỗi thời gian có chiều dài n ($m << n$) chúng tôi sử dụng như những định nghĩa được trình bày trong [11].

III. PHƯƠNG PHÁP ĐỀ XUẤT

Trong phần này chúng tôi trình bày giải thuật tìm kiếm motif xấp xỉ trên dữ liệu chuỗi thời gian do chúng tôi đề xuất. Các thuật toán trong phần này được thiết kế để tìm kiếm motif trên dữ liệu chuỗi thời gian được xác định trong định nghĩa 1. Các bài toán tìm kiếm k motifs, cụm motif và tìm kiếm motif trong trường hợp chuỗi con có thể được giải quyết một cách tương tự.

Ý tưởng cơ bản của thuật toán này là sử dụng cấu trúc R*-tree để tìm kiếm lân cận gần nhất. Với mỗi chuỗi thời gian có chiều dài n trong CSDL, chúng tôi tạo một vùng bao MBR trong không gian m chiều ($m << n$). Sau đó, TSD được chèn vào R*-tree dựa vào vùng bao MBR của nó.

Để tìm lân cận gần nhất của một chuỗi s bằng cách tìm trên R*-tree, chúng ta cần một hàm tính khoảng cách $D_{region}(s, R)$ giữa chuỗi s và vùng bao MBR R kết hợp với một nút trong cấu trúc chỉ mục sao cho $D_{region}(s, R) \leq D(s, C)$, "C được bao trong *MBR R*.

- **Định nghĩa 4.** Cho một chuỗi thời gian s có chiều dài n, một tập các chuỗi thời gian C và vùng bao MBR R tương ứng của C trong không gian m chiều ($m << n$), i.e., $R = \{R_1, R_2, \ldots, R_m\}$,

Trong đó $R_j = \{(x_{jmin}, y_{jmin}), (x_{jmax}, y_{jmax})\}$ là vùng bao chữ nhật nhỏ nhất trong không gian hai chiều.

Hàm tính khoảng cách $D_{region}(s, R)$ giữa chuỗi s và MBR R được định nghĩa là:

$$D_{region}(s, R) = \sqrt{\sum_{j=1}^{m} D_{region_j}(s_j, R_j)}$$

Trong đó,

$$D_{region_j}(s_j, R_j) = \sum_{i=1}^{N} d(s_j, R_j)$$

$$d(s_{j_i}, R_j) = \begin{cases} (y_{j\min} - s_{j_i})^2 & \text{if } s_{ji} < y_{jmin} \\ (s_{j_i} - y_{j\max})^2 & \text{if } s_{ji} > y_{jmax} \\ 0 & \text{otherwise} \end{cases}$$

N là chiều dài của đoạn thứ j.

Bổ đề. $D_{region}(s, R) \leq D(s, C)$, $\forall C$ được bao trong MBR R.

Với $D(s,C) = \sqrt{\sum_{i=1}^{n}(s_i - c_i)^2} = \sqrt{\sum_{j=1}^{m}\sum_{i=1}^{N}(s_j - c_j)^2}$

Chứng minh:

Theo định nghĩa của MBR của một nút U trong cấu trúc chỉ mục và định nghĩa của hàm $D_{region}(s, R)$, Với bất kỳ chuỗi C nào thuộc nút U và vùng bao MBR R của nút U,

ta có
$y_{jmin} \le c_{ji} \le y_{jmax}$, $\forall i = 1, .., N, \forall j = 1, .., m$

Nghĩa là $\forall j = 1, .., m$

$D_{region_j}(s_j, R_j) \le D(s_j, C_j)$

trong đó

$$D(s_j, C_j) = \sum_{i=1}^{N} (s_{ji} - c_{ji})^2$$

vì vậy
$D_{region}(s, R) \le D(s, C)$, $\forall$C thuộc MBR *R*.
Với mỗi chuỗi thời gian S_i trong CSDL chúng ta cần tìm chuỗi lân cận gần nhất với nó trong các chuỗi đã được xem xét trước đó bằng cách sử dụng R*-tree. Khi tìm tới một phần tử trong nút lá, chuỗi gốc tương ứng với nó, S_j, được khôi phục. *Distance*(S_i, S_j) được ghi lại như khoảng cách tốt nhất đến thời điểm hiện tại (*best-so-far*) và vị trí của hai chuỗi S_i, S_j được ghi lại như cặp ứng viên motif tốt nhất tính đến thời điểm hiện tại. Sau vòng lặp kế, giả sử S_x, S_y là cặp chuỗi tương tự nhất, nếu *Distance*(S_x, S_y) < *best-so-far* thì *Distance*(S_x, S_y) được ghi lại để thay thế cho *best-so-far* hiện tại và vị trí của S_x, S_y được ghi lại như là cặp ứng viên motif tốt nhất. Tiến trình được lặp lại cho đến khi không còn chuỗi nào cần được xem xét. *Distance*(S_i, S_j) được dùng trong bài báo này là khoảng cách euclid. Hình 5 trình bày thuật toán tìm kiếm cặp motif xấp xỉ của chúng tôi với sự trợ giúp của R*-tree dựa trên MBRs.

Thuật toán: Tìm kiếm cặp motif xấp xỉ với sự trợ giúp của R*-tree dựa trên MBRs

---

//*S* là CSDL TSD

Procedure (I_1, I_2) = Finding_Motif(*S*)
BestSoFar_distance = INF
For *j* = 1 to *n*
  Find MBR$_j$ of d_j.
  If (R*-tree != null)
    x = Nearest neighbor(*j*, R*-tree)
    if (Distance(s_j, s_x) < BestSoFar_Distance)
    BestSoFar_Distance = Distance(s_j, s_x)
      $I_1 = j$
      $I_2 = x$
    Add(MBRj, R*-tree)

//Tìm lân cận gần nhất của chuỗi j
Nearest neighbor(j, R*-tree)
  Duyệt R*-tree bắt đầu từ nút gốc
  Tìm nút là m có MBR gần nhất với dj
  For i = 1 to số phần tử trong m
  Tìm phần tử y có MBR gần nhất với dj
  Return y

//Chèn TSD j vào R*-tree dựa vào MBRj

Add(MBR$_j$, R*-tree)
  Chọn cây con sao có MBRs cần được mở rộng ít nhất.
  Thêm phần tử mới.
 Nếu nút lá đầy
  Tách nút theo tiêu chuẩn: tối thiểu hóa diện tích của hai vùng bao của hai nút sau khi tách.
  Tiến trình tách nút được lặp lại cho các nút cha nếu nút cha đầy do việc tách nút con.

Hình 5. Thuật toán tìm kiếm motif xấp xỉ bằng R*-tree dựa trên MBRs.

IV. KẾT QUẢ THỰC NGHIỆM

Chúng tôi thực nghiệm so sánh thời gian chạy và độ hiệu quả của thuật toán đề xuất so với phương pháp được sử dụng phổ biến là phép chiếu ngẫu nhiên. Giải thuật chiếu ngẫu nhiên được lựa chọn để so sánh vì thuật toán này đã được sử dụng rộng rãi để phát hiện motif trên chuỗi thời gian từ khi nó được giới thiệu, nó có thể phát hiện motif trong thời gian tuyến tính, đây cũng là thuật toán được trích dẫn nhiều và là cơ sở cho nhiều cách tiếp cận hiện nay cho bài toán phát hiện motif trong dữ liệu chuỗi thời gian.

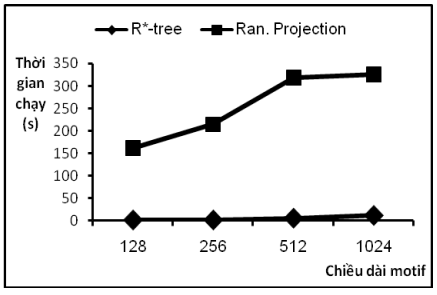
Các giải thuật dùng trong thực nghiệm được viết bằng ngôn ngữ *C#* và chạy trên máy *Core 2 Duo 1.60 GHz, 1.00 GB RAM*. Độ hiệu quả của thuật toán là số lần gọi hàm tính

khoảng cách Euclid của phương pháp được áp dụng chia cho số lần gọi hàm của thuật toán brute-force. Miền giá trị của độ hiệu quả thuộc [0,1]. Độ hiệu quả của phương pháp càng nhỏ thì phương pháp càng hiệu quả.

Trong thực nghiệm chúng tôi sử dụng các tập dữ liệu thực thuộc ba lĩnh vực khác nhau được lấy từ nhiều nguồn khác nhau đã được công bố trên internet là: Stock, ECG, Consumer and Waveform. Thực nghiệm được thực hiện trên các chiều dài motif khác nhau (128 – 1024), kích thước dữ liệu khác nhau (10.000 – 30.000). Với phương pháp dùng R*-tree chúng tôi xây dựng vùng bao MBR bao các chuỗi thời gian theo tỉ lệ 32:1. Với phép chiếu chúng tôi thu giảm số chiều theo tỉ lệ trên và biến đổi thành từ SAX có bộ ký tự là 5, số vị trí được dùng làm mặt nạ che được chọn ngẫu nhiên từ 2 đến 20 để đảm bảo sự phân bố của phép chiếu đủ rộng nhằm tránh độ phức tạp của phép chiếu tăng thành bậc hai.

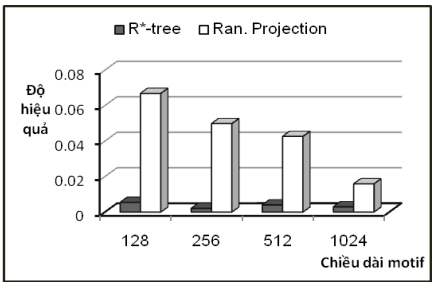
Do giới hạn của bài báo, chúng tôi chỉ trình bày một số kết quả thực nghiệm tiêu biểu.

Hình 6 trình bày kết quả thực nghiệm về thời gian thực hiện của hai giải thuật trên tập dữ liệu Stock với chiều dài motif khác nhau. Số chuỗi được chọn cố định là 10000.



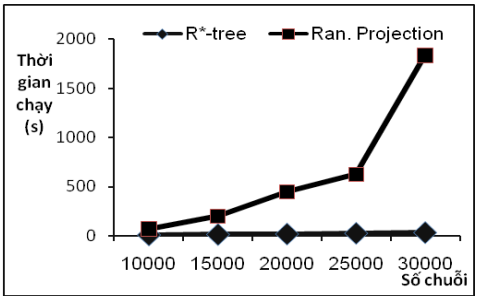
Hình 6. Thời gian thực hiện của hai giải thuật trên tập dữ liệu Stock với chiều dài motif khác nhau

Hình 7 trình bày kết quả thực nghiệm về độ hiệu quả của hai giải thuật thực nghiệm trên tập dữ liệu Stock với chiều dài motif khác nhau. Số chuỗi được chọn cố định là 10000.



Hình 7. Độ hiệu quả của hai giải thuật. Thực nghiệm trên tập dữ liệu Stock với chiều dài motif khác nhau.

Hình 8 trình bày kết quả thực nghiệm về thời gian chạy của hai giải thuật trên tập dữ liệu Stock với kích thước dữ liệu khác nhau chiều dài motif được chọn cố định là 512.



Hình 8. Thời gian thực hiện của hai giải thuật trên tập dữ liệu Stock với kích thước dữ liệu khác nhau.

Kết quả thực nghiệm trên các tập dữ liệu thực cho thấy phương pháp đề xuất hiệu quả hơn so với phép chiếu ngẫu nhiên về cả hai mặt thời gian thực hiện và độ hiệu quả.

V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong bài báo này chúng tôi giới thiệu một phương pháp mới để tìm kiếm motif xấp xỉ trên dữ liệu chuỗi thời gian. Phương pháp này thực hiện phân tích trực tiếp trên dữ liệu số mà không cần phải qua giai đoạn biến đổi thành chuỗi ký tự như các phương pháp tìm kiếm motif xấp xỉ đã công bố trước đây. Phương pháp này hiệu quả về mặt thời gian vì nó chỉ cần một lần quét qua toàn bộ CSDL và một vài lần truy cập chuỗi gốc để thẩm định kết quả. Nó cũng hiệu quả về mặt không

gian vì chỉ có thông tin về vùng bao MBR của chuỗi thời gian được lưu trong bộ nhớ. Kết quả thực nghiệm với các tập dữ liệu nêu ở phần 4 cho thấy phương pháp do chúng tôi đề xuất hiệu quả hơn so với phương pháp thông dụng là phép chiếu ngẫu nhiên về cả hai mặt: thời gian thực thi và độ hiệu quả.

Giới hạn của phương pháp dựa vào R*-tree là R*-tree có thể không thực hiện tốt với dữ liệu chuỗi thời gian có số chiều cao. Chúng tôi sẽ tiếp tục nghiên cứu thay thế R*-tree trong phương pháp của chúng tôi bằng một cấu trúc chỉ mục đa chiều thích ứng tốt hơn với dữ liệu có số chiều cao, chẳng hạn như chỉ mục đường chân trời [9].

TÀI LIỆU THAM KHẢO

[1] N. Beckmann, H. Kriegel, R. Schneider, B. Seeger, "The R*-tree: An efficient and robust access method for points and rectangles," in *Proc. of 1990 ACM SIGMOD Conf.*, Atlantic City, NJ, 1990, pp. 322-331.

[2] S. S. Cheung, and T. P. Nguyen, (2005). *Mining Arbitrary-Length Repeated Patterns in Television Broadcast*. ICIP (3) 2005: 181- 184.

[3] B. Chiu, E. Keogh, and S. Lonardi, *Probabilistic discovery of time series motifs*, Proc. of the 9th International Conference on Knowledge Discovery and Data mining (KDD'03), pp. 493–498, 2003.

[4] N. Castro and P. Azevedo, "Multiresolution Motif Discovery in Time Series," in *Proceedings of the SIAM International Conference on Data Mining (SDM 2010)*, Columbus, Ohio, USA, 2010, pp. 665-676.

[5] F. Erich, G.Thiemo, N. Jiri, S. Bernhard, (2009). *On-line motif detection in time series with SwiftMotif*. In: Pattern Recognition 42(11):3015-3031. Elsevier.

[6] P. Ferreira, P. Azevedo, C. Silva, and R. Brito, (2006). *Mining approximate motifs in time series*, in Proceedings of the 9th International Conference on Discovery Science, pp. 89-101.

[7] A. Guttman, "R-trees: a Dynamic Index Structure for Spatial Searching," in *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, 1984, pp. 47-57.

[8] J. Lin, E. Keogh, S. Lonardi, P. Patel, (2002). *Finding motifs in time series*. In: Proc. 2nd Workshop on Temporal Data Mining. Edmonton, Alberta, Canada.

[9] Li, Q., Lopez, I.F.V. and Moon, B.: Skyline Index for time series data, IEEE Trans. on Knowledge and Data Engineering, Vol. 16, No. 4 (2004)

[10] D. Minnen, C. Isbell, I. Essa, and T. Starner, (2007). *Detecting Subdimensional Motifs: An Effcient Algorithm for Generalized Multivariate Pattern Discovery*, Seventh IEEE International Conference on Data Mining, pp 601-606.

[11] A. Mueen, E. Keogh, Q. Zhu, S. Cash, and B. West-over, (2009). *Exact Discovery of Time Series Motifs*, in the Proceedings of SIAM International Conference on Data Mining, pp. 473-484.

[12] A. Metwally, D. Agrawal, A. El Abbadi , "Efficient Computation of Frequent and Top-k Elements in Data Streams," in *Proceedings of the 10th International Conference on Database Theory*, 2005, pp. 398-412.

[13] T. Oates (2002). PERUSE: *An Unsupervised Algorithm for Finding Recurring Patterns in Time Series*, Second IEEE International Conference on Data Mining, pp. 330.

[14] J. Shieh and E. Keogh, "iSAX: indexing and mining terabyte sized time series," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery*

and Data Mining, 2008, pp. 623-631.

[15] Y. Tanaka, K. Iwamoto, and K. Uehara, *Discovery of time-series motif from multi-dimensional data based on MDL principle*, Machine Learning, 58(2-3):269–300, 2005.

[16] L. Xie, (2005). *Unsupervised Pattern Discovery for Multimedia Sequences*. Ph.D. Thesis, Columbia University.

[17] D. Yankov, E. Keogh, J. Medina, B. Chiu, and V. Zordan, (2007). *Detecting Motifs Under Uniform Scaling*, in Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 844-853.