ĐẠI HỌC THÁI NGUYÊN

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

NGUYỄN NGỌC ANH

NGHIÊN CỨU VÀ THỬ NGHIỆM MỘT SỐ THUẬT TOÁN PHÁT HIỆN CÁC ĐỒ THỊ CON THƯỜNG XUYÊN

LUẬN VĂN THẠC SĨ KHOA HỌC MÁY TÍNH

Thái Nguyên - 2014

MỞ ĐẦU

Hiện nay, các phương pháp khai phá dữ liệu đang phải đối diện với vấn đề số lượng ngày càng gia tăng của các đối tượng dữ liệu phức tạp. Bên cạnh đó đồ thị là một cấu trúc dữ liệu tổng quát, có thể sử dụng để mô hình hóa các đối dữ liệu tượng phức tạp đó và vấn đề khai phá đồ thị con thường xuyên là một trong những vấn đề quan trọng trong khai phá đồ thị. Việc khai phá đồ thị để tìm đồ thị con thường xuyên nhằm xác định tất cả các đồ thị con trong một tập dữ liệu đồ thị với giá trị ngưỡng cho trước [1],[3].

Những khó khăn của vấn đề khai phá đồ thị con thường xuyên nảy sinh hai vấn đề, đó là: liệt kê tất cả các đồ thị con trong CSDL đồ thị và tính toán hàm hỗ trợ của các đồ thị con này trong CSDL. Do các đỉnh của đồ thị có thể được sắp xếp theo nhiều cách, một đồ thị có thể có số lượng lớn các bản sao hình học tương đương, được gọi là đồ thị đẳng cấu. Để liệt kê tất cả các đồ thị con, ta phải tính toán phù hợp với quy tắc biểu diễn đồ thị để giải quyết vấn đề đồ thị đẳng cấu. Hơn nữa, việc kiểm tra nếu một đồ thị có chứa trong một CSDL đồ thị hay không được xem như bài toán NP-khó và được gọi là bài toán đồ thị con đẳng cấu. Trong tất cả các trường hợp, việc tính toán hàm hỗ trợ chiếm chi phí nhiều nhất trong việc tìm các đồ thị con thường xuyên của CSDL. Tuy nhiên, sự phức tạp của những vấn đề này sẽ giảm khi CSDL đồ thị có thêm thông tin về các đỉnh và các cạnh đã được gán nhãn. Có thể sử dụng các nhãn để hạn chế các đỉnh có thể tạo thành các cặp trong quá trình kiểm tra sự đẳng cấu của đồ thị con. Tuy nhiên, nếu CSDL đồ thị chưa được gán nhãn hoặc chỉ có một số ít các nhãn thì độ phức tạp của bài toán sẽ làm giảm đáng kể kích thước của tập dữ liệu.

Như vậy, vấn đề khai phá đồ thị nói chung và khai phá đồ thị con thường xuyên nói riêng cũng gặp nhiều khó khăn, vì vậy ta cần lựa chọn phương pháp

và thuật toán phù hợp để giải quyết cho từng bài toán cụ thể, đem lại hiệu quả cao đó chính là ý nghĩa thực tiễn của đề tài.

❖ Nội dung của luận văn và các vấn đề cần giải quyết:

1. Tìm hiểu về các phương pháp khai phá dữ liệu đồ thị.
2. Tìm hiểu các thuật toán phát hiện đồ thị con thường xuyên trong CSDL đồ thị.
3. Cài đặt thử nghiệm thuật toán phát hiện các đồ thị con thường xuyên trong CSDL đồ thị

❖ Phương pháp nghiên cứu

+ Nghiên cứu về khai phá dữ liệu đồ thị với trọng tâm là phát hiện các đồ thị con thường xuyên trong CSDL đồ thị.

+ Tìm hiểu các nguồn thông tin từ các sách,bài báo,tạp chí, Internet..,liên quan đến khai phá dữ liệu đồ thị.

❖ Cấu trúc luận văn chia làm 4 chương:

Chương 1: " **Tổng quan về khai phá dữ liệu đồ thị** " trình bày tổng quan các hướng nghiên cứu hiện nay về khai phá dữ liệu đồ thị.

Chương 2: " **Phát hiện các cấu trúc con thường xuyên** " trình bày cơ sở lý thuyết đồ thị, cách tiếp cận dựa trên Apriori, cách tiếp cận dựa trên sự phát triển mẫu.

Chương 3: " **Các thuật toán phát hiện đồ thị con thường xuyên** " trình bày một số thuật toán phát hiện đồ thị con thường xuyên theo chiến lược tìm kiếm theo chiều rộng và chiều sâu.

Chương 4: " **Thiết kế hệ thống thử nghiệm** " trình bày kết quả cài đặt của thuật toán trong chương 3.

CHƯƠNG 1: TỔNG QUAN VỀ KHAI PHÁ DỮ LIỆU ĐỒ THỊ

1.1. TỔNG QUAN VỀ KHAI PHÁ DỮ LIỆU ĐỒ THỊ:

Khai phá dữ liệu đồ thị là một trong số các lĩnh vực quan trọng trong khai phá dữ liệu. Hầu hết nguồn dữ liệu hiện nay có thể biểu diễn được dưới dạng cấu trúc dữ liệu đồ thị, chẳng hạn như: dữ liệu từ mạng Internet, mạng xã hội, cấu trúc protein, hợp chất hóa học,... Do đó, khai phá dữ liệu đồ thị nhằm tìm kiếm các thông tin hữu ích trong một lượng lớn dữ liệu là vấn đề đang được các nhà nghiên cứu và các tổ chức CNTT quan tâm.

1.1.1. Định nghĩa dữ liệu lớn:

Hiện nay, thuật ngữ "Dữ liệu lớn" (Big data) đang thu hút sự quan tâm cũng như đặt ra những thách thức mới với các nhà nghiên cứu, các nhà cung cấp dịch vụ công nghệ thông tin và các tổ chức, doanh nghiệp. Dữ liệu lớn được xem như sự ra đời tất yếu của quá trình bùng nổ thông tin.

Trong nhiều năm qua, các doanh nghiệp thường đưa ra các quyết định kinh doanh dựa trên dữ liệu giao dịch được lưu trữ trong cơ sở dữ liệu quan hệ. Ngoài ra những dữ liệu quan trọng lại thường ở dạng tiềm năng, phi truyền thống, phi cấu trúc lại có thể được khai thác một cách hữu ích, giảm chi phí cả về lưu trữ và tính toán. Khi dữ liệu lớn được được khai thác và phân tích, kết hợp với dữ liệu doanh nghiệp truyền thống thì các doanh nghiệp sẽ có cái nhìn toàn diện và sâu sắc hơn về tình hình kinh doanh của họ, dẫn tới nâng cao năng suất và vị thế cạnh tranh. Do đó, ngày càng có nhiều công ty tìm kiếm để có được các dữ liệu phi truyền thống nhưng rất có giá trị trong công việc kinh doanh này.

Có thể định nghĩa một cách chung nhất thì "Dữ liệu lớn" là một tập hợp của các tập dữ liệu lớn và/hoặc phức tạp mà những phương pháp hiện tại của CNTT chưa thể phân tích và xử lý tốt được chúng.

Dữ liệu lớn bao gồm cả tính chất về độ lớn lưu trữ (Volume), đa dạng, phức tạp (Variety) và tăng trưởng nhanh chóng (Velocity)[8].

Dữ liệu lớn thường đề cập tới các kiểu dữ liệu như sau:

- Dữ liệu doanh nghiệp truyền thống: bao gồm các thông tin khách hàng, dữ liệu giao dịch, dữ liệu kế toán tổng hợp.
- Dữ liệu cảm biến hoặc máy sinh dữ liệu: bao gồm các bản ghi chi tiết các cuộc gọi, nhật ký web, hệ đo thông minh, dữ liệu từ các cảm biến, các hệ thống dữ liệu truyền thống.
- Dữ liệu xã hội: bao gồm các luồng thông tin phản hồi của khách hàng, dữ liệu từ các trang nhật ký và mạng xã hội như Twitter, Facebook, ...

1.1.2. Giải pháp dữ liệu lớn của một số nhà cung cấp dịch vụ:

* *Giải pháp Big data của Oracle*

Oracle là nhà cung cấp đầu tiên cung cấp một giải pháp hoàn chỉnh và tích hợp để giải quyết đầy đủ yêu cầu về dữ liệu lớn của doanh nghiệp. Các dữ liệu lớn của Oracle tập trung trên ý tưởng có thể phát triển kiến trúc dữ liệu doanh nghiệp hiện tại để kết hợp dữ liệu lớn và cung cấp giá trị kinh doanh, linh hoạt, hiệu suất để giải quyết yêu cầu về dữ liệu lớn với doanh nghiệp.

Với việc giới thiệu ứng dụng Quản lý Dữ liệu lớn (Oracle Big Data Appliance), Oracle cung cấp một giải pháp hoàn chỉnh đáp ứng mọi yêu cầu liên quan đến dữ liệu lớn của doanh nghiệp. Thiết bị xử lý dữ liệu lớn Oracle Big Data Appliance, cùng với máy chủ cơ sở dữ liệu Oracle Exadata và Máy chủ thông tin hỗ trợ ra quyết định Oracle Exalytics mới, giúp khách hàng có để thu thập, tổ chức, phân tích và khai thác tối đa giá trị của dữ liệu lớn.

Oracle Big Data Appliance có thể được tích hợp dễ dàng với cơ sở dữ liệu Oracle Database 11g, Oracle Exadata Database Machine và Oracle Exalytics Business Intelligence Machine.

** Giải pháp Big Data của Microsoft*

Giải pháp Big Data của Microsoft dựa trên nền tảng SQL Server, Hadoop, Windows Azure và Windows Server, cung cấp các công cụ quản lý, mở rộng nhằm đạt được cái nhìn sâu sắc hơn về dữ liệu của doanh nghiệp, thúc đẩy hiệu quả kinh doanh.

Microsoft Big Data cho phép quản lý hầu như bất kỳ loại dữ liệu nào, bất kể kích thước hoặc vị trí. Microsoft sử dụng SQL Server 2012 và SQL Server Parallel Data Warehouse để quản lý các dữ liệu lớn có cấu trúc. Với dữ liệu phi cấu trúc, Microsoft sử dụng Hadoop trên Windows Azure và Windows Server, sẽ cho phép xử lý dữ liệu phi cấu trúc với quy mô hàng petabyte. Với dữ liệu luồng, Microsoft sử dụng công cụ SQL Server StreamInsight để quản lý các dữ liệu luồng với thời gian thực.

Microsoft Big Data cho phép làm phong phú thêm dữ liệu với bất kỳ loại dữ liệu nào: Cửa hàng dữ liệu Azure Marketplace cho phép các doanh nghiệp có được dữ liệu của bên thứ ba; bộ công cụ phòng thí nghiệm Data Explorer Azure dành cho các tập dữ liệu đề xuất và Data Hub dành cho việc tạo ra các cửa hàng dữ liệu riêng.

1.2. TỔNG QUAN VỀ KHAI PHÁ ĐỒ THỊ CON THƯỜNG XUYÊN:

Cho một CSDL đồ thị D, một hàm hỗ trợ của đồ thị G trong D, được viết là *sup(G, D)* là số lượng các đồ thị trong D có chứa đồ thị G như một cạnh tạo nên đồ thị con. Cho giá trị ngưỡng hỗ trợ cực tiểu *smin*, vấn đề khai phá đồ thị con thường xuyên bao gồm việc tìm ra các đồ thị liên thông thường xuyên trong D.

Có hai nhóm phương pháp được đề xuất để giải quyết vấn đề trên, đó là: nhóm phương pháp khai phá theo chiều rộng và nhóm phương pháp khai phá theo chiều sâu:

Một số kỹ thuật khai phá theo chiều rộng như: kỹ thuật AGM được phát triển bởi Inokuchi, kỹ thuật FSG được đề xuất bởi Kuramochi và Karypis. Các kỹ thuật này khai phá đồ thị theo từng mức trong đó mỗi mức chứa các đồ thị có nhiều hơn một đỉnh hoặc một cạnh so với mức trước đó. Các đồ thị thường xuyên của mức tiếp theo được tìm ra bằng cách, đầu tiên tạo ra các đồ thị ứng viên với các cặp đồ thị của mức hiện tại, sau đó lọc ra các đồ thị không thường xuyên. Ưu điểm chính của những kỹ thuật này dựa trên nguyên tắc ưu tiên bằng cách một đồ thị chỉ được xem là thường xuyên nếu tất cả các đồ thị con của nó là thường xuyên. Vì một đồ thị được tìm ra sau khi tìm ra các đồ thị con của nó, do đó có thể loại bỏ các đồ thị không thường xuyên mà không cần phải tính toán hàm hỗ trợ của chúng bằng cách kiểm tra nếu các đồ thị con của chúng là thường xuyên. Tuy nhiên, nhóm phương pháp tìm kiếm theo chiều rộng có hai vấn đề đó là: sinh ra nhiều đồ thị ứng viên và yêu cầu về lưu trữ các đồ thị thường xuyên ở mỗi mức.

Nhóm phương pháp khai phá theo chiều sâu đã khắc phục những vấn đề này bằng cách tìm kiếm đồ thị theo chiều sâu, có thể kể đến một số thuật toán như: gSpan được đề xuất bởi Han và Yan, FFSM được đề xuất bởi Huan, và GASTON bởi Nijssen và Kok. Tư tưởng của nhóm phương pháp này bắt đầu với một đồ thị có chứa một đỉnh hoặc một cạnh thường xuyên, những kỹ thuật này được mở rộng đệ quy bằng cách thêm mới một cạnh giữa hai đỉnh hiện tại hoặc thêm mới một đỉnh kết nối tới một đỉnh hiện tại khác. Vì một đồ thị là không thường xuyên hơn các đồ thị con của nó, do đó không cần mở rộng tới các đồ thị không thường xuyên. Các đồ thị không thường xuyên có thể được bỏ bớt mà không xảy ra rủi ro gì trong quá trình khai phá.

1.3. KẾT LUẬN

Chương 1 trình bày tổng quan về khai phá dữ liệu đồ thị trong đó có nêu vấn đề của khai phá dữ liệu đồ thị là tìm những thông tin hữu ích trong một

lượng lớn dữ liệu, đưa ra định nghĩa chung nhất về dữ liệu lớn (Big Data) và các giải pháp Big Data của Oracle và Microsoft.

Trình bày tổng quan về khai phá đồ thị con thường xuyên theo hai nhóm phương pháp đó là nhóm phương pháp khai phá theo chiều rộng và nhóm phương pháp khai phá theo chiều sâu cùng với ưu và nhược điểm của hai nhóm phương pháp này.

CHƯƠNG 2: PHÁT HIỆN CÁC CẤU TRÚC CON THƯỜNG XUYÊN

2.1. CƠ SỞ LÝ THUYẾT ĐỒ THỊ

Chúng ta biểu diễn tập đỉnh của đồ thị g bằng $V(g)$ và tập cạnh bằng $E(g)$. Một hàm nhãn L ánh xạ một đỉnh hoặc một cạnh tới một nhãn. Một đồ thị $g$là một đồ thị con của đồ thị g' khác nếu tồn tại một đồ thị con đẳng cấu từ g tới g'.

2.1.1. Định nghĩa 2.1 (Graph):

Chomột nhãn node bằng chữ cái (alphabet) L_V và một nhãn cạnh bằng chữ cái L_E đồ thị g (có hướng) được định nghĩa bằng bộ gồm 4 thành phần $g=(V, E, \mu, \nu)$, trong đó:

- V biểu diễn một tập hữu hạn các node.
- $E \subseteq V \times V$ biểu diễn một tập các cạnh.
- $\mu: V \rightarrow L_V$ biểu diễn một hàm ghi nhãn node.
- $\nu: E \rightarrow L_E$ biểu diễn một hàm ghi nhãn cạnh.

Tập V có thể được coi là một tập các định danh nút và thường được chọn bằng $V = \{1, ..., | V |\}$. Trong khi V xác định các nút, tập các cạnh E thể hiện cấu trúc của đồ thị. Đó là một nút $u \in V$ được kết nối với một nút $v \in V$ bằng một cạnh$= (u, v)$ nếu $(u, v) \in E$. Hàm ghi nhãn có thể được sử dụng để tích hợp thông tin về các node và các cạnh vào trong các đồ thị bằng cách gán các thuộc tính từ L_V và L_E tới các node và các cạnh tương ứng.

Đồ thị được định nghĩa ở trên bao gồm một số trường hợp đặc biệt. Để định nghĩa đồ thị vô hướng, cho một thể hiện yêu cầu $\langle u, v \rangle \in E$ cho mỗi cạnh $\langle u, v \rangle \in E$ sao cho $\nu(u, v) = \nu(v, u)$. Trong trường hợp đồ thị không thuộc tính, bảng chữ cái nhãn được xác định bởi $L_V = L_E = \emptyset$, bởi vậy mỗi node và mỗi

cạnh được gán nhãn *null* nhãn ∅. Đồ thị rỗng được định nghĩa bằng $g_\varepsilon = (\emptyset, \emptyset, \mu_\varepsilon, v_\varepsilon)$.

2.1.2. Định nghĩa 2.2 (Subgraph):

Cho $g_1=(V_1,E_1,\mu_1,v_1)$ và $g_2=(V_2,E_2,\mu_2,v_2)$ là các đồ thị, đồ thị g_1 là một đồ thị con của g_2 , ký hiệu $g_1 \subseteq g_2$ nếu

- $V_1 \subseteq V_2$.
- $E_1 = E_2 \cap (V_1 \times V_1)$.
- $\mu_1(u) = \mu_2(u)$ cho tất cả $u \in V_1$.
- $v_1(u, v) = v_2(u, v)$ cho tất cả $(u, v) \in E_1$.

Ngược lại, đồ thị g_2 được gọi là một đồ thị con của g_1 đôi khi điều kiện thứ hai của định nghĩa này được thay thế bằng $E_1 \subseteq E_2$.

2.1.3. Định nghĩa 2.3 (Graph Isomorphism):

Cho $g_1=(V_1,E_1,\mu_1,v_1)$ và $g_2=(V_2,E_2,\mu_2,v_2)$ là các đồ thị.Một đồ thị đẳng cấu giữa g_1 và g_2 là một hàm song ánh $f: V_1 \rightarrow V_2$ thỏa mãn:

- $\mu_1(u) = \mu_2(f(u))$ cho tất cả các node $u \in V_1$.
- Cho mỗi cạnh $e_1=(u, v) \in E_1$, tồn tại một cạnh $e_2=(f(u), f(v)) \in E_2$ sao cho $v_1(e_1) = v_2(e_2)$.
- Cho mỗi cạnh $e_2=(u, v) \in E_2$, tồn tại một cạnh $e_1=(f^{-1}(u), f^{-1}(v)) \in E_1$ sao cho $v_1(e_1) = v_2(e_2)$.

Hai đồ thị g_1 và g_2 được gọi là đẳng cấu nếu tồn tại một đồ thị đẳng cấu giữa chúng.