Tạp chí Tin học và Điều khiển học, T.25, S.2 (2009), 166–177

PHÂN LOẠI TỰ ĐỘNG CÁC KẾT QUẢ TRUY VẤN SỬ DỤNG CÁC SỞ THÍCH CỦA NGƯỜI DÙNG

NGUYỄN KIM ANH

Khoa Công nghệ thông tin, Đại học Bách khoa Hà Nội

Abstract. Database queries are often exploratory and users often find their queries return too many answers, which are most irrelevant. The existing work either classifies or ranks the results to help users locate interesting results. However, most existing work assumes that all users have the same user preferences, but in the real life different users often have different preferences. This paper proposes an executable solution to automatical classification of the results of SQL queries via exploring user preferences using a query log.

Tóm tắt. Phần lớn các câu truy vấn đối với một cơ sở dữ liệu thường mang tính thăm dò và những người dùng thường nhận được quá nhiều câu trả lời, trong đó nhiều câu trả lời là không thích đáng với câu hỏi của họ. Các công trình hiện nay thường tiến hành phân loại hoặc xếp hạng các kết quả để giúp những người dùng định vị được các kết quả thú vị. Tuy nhiên, hầu hết các công trình này đều giả thiết rằng tất cả những người dùng đều có cùng sở thích, mặc dù, trong đời sống thực, những người dùng khác nhau thường có các sở thích khác nhau. Bài báo này đề nghị một giải pháp khả thi cho phép phân loại tự động các kết quả của các câu truy vấn SQL thông qua việc khai phá các sở thích của người dùng sử dụng một tập các câu truy vấn trong quá khứ.

1. GIỚI THIỆU

Phân loại tự động một tập dữ liệu là quá trình sàng lọc, phân chia tập dữ liệu đó một cách phù hợp và tự động thành các nhóm dữ liệu nhỏ hơn, mỗi nhóm dữ liệu hàm chứa một lượng thông tin cụ thể và tương đối phân biệt. Phân loại dữ liệu là hành vi phổ biến trong xã hội và phục vụ cho rất nhiều đối tượng, mục đích khác nhau. Một hệ thống dịch vụ thông minh phải có khả năng thực hiện quá trình đó một cách tự động trên cơ sở hiểu biết về sở thích, mong muốn của khách hàng, giúp họ đưa ra được những quyết định đúng đắn. Với đặc điểm như vậy, một hệ thống phân loại tự động kết quả vừa là một hệ tương tác người-máy linh hoạt và mềm dẻo, vừa là một hệ trợ giúp quyết định.

Trong những năm gần đây, nghiên cứu về vấn đề phân loại tự động đã đạt được nhiều thành tựu rất đáng kể, với sự trợ giúp đắc lực của các kỹ thuật khai phá dữ liệu (học tự động có giám sát/không có giám sát, phân tích cụm, xử lý phân lớp,...). Trong các hệ tìm kiếm thông tin, phân loại tự động cùng với xếp hạng kết quả đã trở thành hai giải pháp chủ lực cho bài toán quá tải thông tin trên Internet. Bên cạnh các hệ thống tìm kiếm sử dụng giải pháp xếp hạng kết quả rất nổi tiếng trên thế giới như Google, Yahoo!, MSN Live Search,...

đã bắt đầu xuất hiện những hệ thống thử nghiệm mô hình phân loại tự động kết quả. Có thể kể ra đây một số ví dụ: Website Webclust.com của đại học Pisa, Italy, hệ thống Clusty.com của nhóm nghiên cứu ở Pittsburgh, (PA, USA) và tập đoàn Vivisimo,...

Trong khi đó, bài toán phân loại tự động trên cơ sở dữ liệu (CSDL) quan hệ - với các đặc thù về cấu trúc và dữ liệu còn là một hướng nghiên cứu rất mới mẻ, và chỉ được tập trung nghiên cứu trong một hai năm trở lại đây với các kết quả không đáng kể. Thực tế, mô hình CSDL quan hệ là một mô hình rất mạnh trong quản lý và xử lý dữ liệu và thông tin, hiện còn được sử dụng rất rộng rãi ở Việt Nam cũng như trên thế giới. Bài báo khai thác một số ý tưởng trong các công trình có liên quan [1,2] để đề xuất một giải pháp đối với vấn đề phân loại tự động kết quả truy vấn trên CSDL quan hệ trên cơ sở cộng tác với người dùng.

Nội dung của bài báo được trình bày trong 5 mục. Mục 1 trình bày về nhu cầu phân loại kết quả truy vấn SQL tự động. Mục 2, các khái niệm cơ bản: giá trị ưa thích, cụm sở thích và cây điều hướng, là các khái niệm lý thuyết nền tảng của bài báo. Mục 3 là giải pháp đối với bài toán phân cụm dữ liệu. Mục 4 là giải thuật sinh cây điều hướng để phân loại kết quả tự động và cuối cùng, Mục 5, các kết luận và các hướng phát triển trong tương lai đối với vấn đề phân loại tự động các kết quả truy vấn đối với một CSDL quan hệ có kích thước rất lớn.

2. CÁC KHÁI NIỆM CƠ BẢN

2.1. Cây điều hướng

Cây điều hướng (navigation tree) là một công cụ biểu diễn trực quan kết quả của quá trình phân loại tự động kết quả truy vấn theo các thuộc tính trong CSDL. Ý tưởng cơ bản của cây điều hướng là: Đầu tiên, các kết quả truy vấn được phân loại thành các nhóm con theo một thuộc tính A nào đó, sau đó, mỗi nhóm con lại tiếp tục được phân loại thành các nhóm con nhỏ hơn theo một thuộc tính B nào đó. Quá trình trên có thể được thực hiện lặp lại nhiều lần cho tới khi người dùng tìm được các kết quả thích đáng đối với mục đích của người dùng.

Sau đây là định nghĩa hình thức cho cây điều hướng. Để đơn giản, xét CSDL chỉ gồm một quan hệ R xác định trên tập thuộc tính $A = \{A_1, A_2, ..., A_n\}$. Ký hiệu CSDL là D. Xét các câu truy vấn SQL Q trên D với kết quả trả về là D_Q thoả mãn các điều kiện sau:

- Q là một câu truy vấn kiểu SP (chọn-chiếu).
- Điều kiện chọn trên Q có dạng $\wedge_i d_k(B_i)$ với $d_k(B_i)$ là một điều kiện đơn trên thuộc tính B_i, $B_i \in A$.
- Các điều kiện đơn $d_k(B_i)$ trên thuộc tính B_i chỉ có một trong các dạng sau:

  $B_i \in K$ với $K \subset Dom(B_i)$ và B_i là thuộc tính định tính.

  $B_i \in I$ với $I = [b_1, b_2); b_1, b_2 \in Dom(B_i)$ và B_i là thuộc tính định lượng.

Định nghĩa 1. [2] Một cây điều hướng T của một kết quả truy vấn D_Q là một bộ ba (V, E, L) với V là tập các nút, E tập các cạnh và L tập các nhãn nếu với mọi nút $v \in V$

thỏa mãn các điều kiện sau:

• Tồn tại duy nhất một nhãn $l(v) \in L$ ứng với v.

• Tồn tại một điều kiện chọn đơn $dk(A_i)$ trên thuộc tính A_i (ta gọi $dk(A_i)$ là điều kiện chọn của v).

• Tồn tại một tập con khác rỗng $N(v) \subset D_Q$ gồm các bộ thỏa mãn tất cả các điều kiện chọn của các nút từ gốc đến $v(N(v)$ được gọi là tập dữ liệu của $v)$.

Tất cả nút con v_j của v (nếu có) đều có các điều kiện chọn trên cùng một thuộc tính (gọi là thuộc tính chia của v) và có các tập dữ liệu $N(v_j)$ lập thành một phân hoạch của $N(v)$.

Từ Định nghĩa 1 cho thấy, một kết quả truy vấn có thể có nhiều cây điều hướng khác nhau. Ngoài ra, trong cây điều hướng, mỗi nút con không phải là lá chỉ có duy nhất một thuộc tính chia. Khái niệm "lá ở đây được hiểu như sau: Nếu người dùng lựa chọn dừng tại một nút nào đó, thì đó là lá. Trong trường hợp người dùng tiếp tục mở rộng cây, những nút nào không thể chọn được thuộc tính chia và các điều kiện chọn cho các nút con thỏa mãn Định nghĩa 1 sẽ là các nút lá của cây. • Để mô phỏng việc duyệt cây điều hướng, ta đơn giản hóa quá trình này như sau:

• Người dùng chỉ duyệt cây theo mô hình top-down.

• Với mỗi nút trên cây, người dùng chỉ lựa chọn một trong các khả năng:

+ Duyệt nhãn: Chọn xem tất cả các nhãn của các nút con (nếu có). Đối với mỗi nút con, người dùng có thể chọn duyệt hoặc bỏ qua.

+ Duyệt bộ: Chọn xem tất cả các bộ trong tập dữ liệu của nút. Để đơn giản ta không xem xét trường hợp người dùng "xem trước" một vài bộ rồi quay lại chọn duyệt nhãn.

+ Bỏ qua.

• Với nút lá, người dùng luôn lựa chọn duyệt bộ.

Ngoài ra chúng ta giả định rằng người dùng không dừng duyệt cây ngay từ đầu, và luôn chọn duyệt nhãn ở gốc.

2.2. Tập truy vấn quá khứ

Quá trình chia nhóm trên cây điều hướng chỉ thực sự hiệu quả khi lựa chọn được thuộc tính chia phù hợp với mong muốn của người dùng. Do vậy, việc lựa chọn thuộc tính nào để chia nhóm đòi hỏi phải có hiểu biết nhất định về mong muốn hay sở thích của người dùng. Để khai phá được sở thích của người dùng, có thể sử dụng tập các truy vấn trong quá khứ (Query Log trong [1]) của những người sử dụng đã truy cập vào hệ thống trước đây. Thông tin về sở thích của đa số người dùng trước đây có thể được sử dụng để đáp ứng tốt hơn các nhu cầu truy vấn trong tương lai, đặc biệt có thể trợ giúp cho những người sử dụng mới còn chưa có nhiều hiểu biết về CSDL cũng như những tri thức tiềm ẩn trong hệ thống. Chức năng thu thập tập các truy vấn quá khứ đã được hỗ trợ trong hầu hết các hệ quản trị CSDL hiện tại.

2.3. Tập cụm sở thích

Khi người dùng đệ trình một truy vấn đối với một CSDL cụ thể, những mong muốn và sở thích của người dùng được phản ánh thông qua các điều kiện tra cứu và những thông tin cần tra cứu. Do vậy, việc phân tích các điều kiện trong câu truy vấn của người dùng và kết quả trả về của câu truy vấn có thể phát hiện được những mong muốn và sở thích của người dùng. Sau đây, sẽ đưa ra định nghĩa hình thức cho khái niệm cụm sở thích của người dùng.

Định nghĩa 2. Cho một quan hệ R trên CSDL D và tập H gồm tất cả các câu truy vấn SP đã được thực hiện trên R. Một cụm sở thích trên R tương ứng với H là một tập con lớn nhất các bộ của R thuộc các kết quả của cùng một tập câu truy vấn trong H.

Mức độ quan tâm của người dùng tới một cụm sở thích C có thể được tính bằng xác suất thực hiện truy vấn trên cụm C, gọi tắt là xác suất truy vấn C. Cụm $C-$ chứa các bộ không xuất hiện trong bất kỳ truy vấn nào sẽ có xác suất truy vấn bằng 0. Dễ thấy các cụm sở thích là rời nhau. Do vậy, tập các cụm sở thích tạo thành một phân hoạch trên quan hệ R.

Định nghĩa 3. Cho một CSDL D với k cụm sở thích $C_1, C_2, ..., C_k$. Xác suất truy vấn cụm C_j với $j = 1, k$, ký hiệu $p(C_j)$, là đại lượng được tính bằng tổng xác suất của các câu truy vấn có kết quả chứa C_j.

Ở đây, xác suất đặt ra một truy vấn được tính bằng tần suất xuất hiện truy vấn trong tập các truy vấn quá khứ.

2.4. Giá trị ưa thích

Với mỗi thuộc tính định lượng A, giả sử $Dom(A) = [v_{\min}, v_{\max})$. Xét tập các truy vấn Q trong H có chứa một điều kiện đơn trên thuộc tính A. Giả sử $DK(A) = \{[a_{11}, a_{12}), ...[a_{m1}, a_{m2})\}$ là các khoảng xuất hiện trong các điều kiện đơn trên thuộc tính A của các truy vấn này. Sắp xếp tập các giá trị $a_{11}, a_{12}, .., a_{m1}, a_{m2}, v_{\min}, v_{\max}$ theo thứ tự tăng dần. Giả sử tập các giá trị sau sắp xếp là $b_1, b_2, .., b_l$. Ta gọi $[b_i, b_{i+1})$ với $i = 1, l-1$ là một khoảng ưa thích đối với A tương ứng với H (gọi tắt là khoảng ưa thích đối với A) nếu $[b_i, b_{i+1})$ được chứa trong một khoảng nào đó của DK. Kí hiệu $p^-(A)$ là hợp của tập tất cả các khoảng không được ưa thích. Kí hiệu $I(A) = \{I_i / I_i$ là một khoảng ưa thích đối với A, $i = 1, l\}$. Dễ thấy, mỗi khoảng ưa thích là một khoảng con lớn nhất đối với A được chứa trong các điều kiện đơn trên thuộc tính A của cùng một tập câu truy vấn trong H. Hơn nữa, $I(A) \cup p^-(A)$ là một phân hoạch đối với $Dom(A)$.

Với mỗi thuộc tính định tính A, với $Dom(A)$ bao gồm một tập hữu hạn các giá trị rời rạc. Xét tập các truy vấn Q trong H có chứa một điều kiện đơn trên thuộc tính A. Một tập giá trị ưa thích K đối với A tương ứng với H (gọi tắt là tập giá trị ưa thích đối với A) là một tập con lớn nhất các giá trị đối với A được chứa trong các điều kiện đơn trên thuộc tính A của cùng một tập câu truy vấn trong H. Kí hiệu $p^-(A)$ là tập tất cả các giá trị không được ưa thích đối với A. Kí hiệu $K(A) = \{K_i / K_i$ là một tập giá trị ưa thích đối với A, $i = 1, l\}$.

Dễ thấy $K(A) \cup p^{-}(A)$ là một phân hoạch đối với $Dom(A)$.

Ví dụ: $H = \{$select * from R where B in $\{a, b, c\}$ and $A = 4$ and $A < 8$, select * from R where $A = 6$ and $A < 10$ and B in $\{a, c\}$, select* from R where B in $\{a, c, d, e\}$ and $A < 15\}$ với $Dom(A) = [1, 20)$ và $Dom(B) = \{a, b, c, d, e, f\}$. Khi đó $I(A) = \{[4, 6), [6, 8), [8, 10), [10, 15)\}$ và $K(A) = \{\{b\}, \{a, c\}, \{d, e\}\}$.

3. PHÂN CỤM DỮ LIỆU

Trong thực tế, tại thời điểm phân cụm, các kết quả truy vấn cũ sẽ được tính toán lại trên dữ liệu mới vì người dùng luôn muốn các thông tin được cập nhật mới nhất. Do vậy, chỉ cần lưu bản thân các câu truy vấn đó chứ không cần lưu kết quả của các câu truy vấn này. Phần này sẽ khai thác các điều kiện của các truy vấn quá khứ.

Thông thường, số lượng các câu truy vấn trong H là cực lớn, do đó, số lượng các cụm sở thích cũng có thể rất lớn và hơn nữa, kích cỡ của các cụm có thể rất nhỏ (thậm chí một số cụm chỉ có một bộ). Điều này khiến việc phân cụm trở nên vô nghĩa. Với mong muốn nâng cao hiệu năng hệ thống và xác định được các cụm sở thích thực sự có nghĩa-phản ánh được các xu hướng hay các chủ đề quan tâm của đa số người dùng-sử dụng một số heuristic tiền xử lý tập các truy vấn quá khứ.

• Loại bỏ đi các truy vấn tồi trong H thông qua một thủ tục "tỉa tập truy vấn" (query pruning). Các truy vấn tồi là các truy vấn có kết quả bằng rỗng, các truy vấn quá chung chung (không có điều kiện trong truy vấn hoặc điều kiện quá tổng quát) hoặc các truy vấn có xác suất quá bé.

• Nới lỏng một số điều kiện quá chặt hay quá cụ thể, quá chi tiết trong các truy vấn thông qua việc khai thác các phân cấp khái niệm đối với các thuộc tính [3]. Sau nới lỏng, một số truy vấn có thể trùng nhau. Do vậy, số lượng các câu truy vấn trong H có thể giảm đi một cách đáng kể.

Sau đây, là các thuật toán cần thiết để xây dựng tập các cụm sở thích.

Thuật toán 1. Tính $K(A)$ đối với thuộc tính định tính

Vào: - Quan hệ R, thuộc tính A của R, $Dom(A)$.

- Tập H các truy vấn trên R. Với mỗi $Q \in H$, $dk(Q)$- ký hiệu điều kiện của truy vấn Q.

Ra: $K(A)$.

Cách tính:

1. Đặt $K = \{K_i / K_i$ là một tập giá trị xuất hiện trong một điều kiện đơn trên thuộc tính A của một truy vấn Q nào đó trong H, $i = 1, m\}$.

2. Tình lực lượng của các K_i, $i = 1, m$.

3. Sắp xếp lại K theo thứ tự tăng dần đối với lực lương của các K_i, $i = 1, m$. Giả sử $K = \{K_1, ..., K_m\}$ sau khi đã sắp xếp.

4. Đặt $K = \{K_1\}$.

5. for each $j = 2, m$:

$i = 1$
while $i < j$ and $K_j \neq \phi$:
if $K_i \subset K_j$ then Xóa K_i trong K_j
if $K_i \cap K_j \not\phi$ then $G = K_i \cap K_j$; Thay K_i trong K bởi G và $K_i - G$;
Xóa G trong K_j
end if
$i = i + 1$
end while
end for
6. Đặt $K(A) = K$.

Mệnh đề 1. *Thuật toán 1 xác định đúng tập $K(A)$ đối với một thuộc tính định tính A.*

Chứng minh. Giả sử $K = \{K_1, ..., K_l\}$. Dễ thấy các tập $K_i,\ i = 1, l$ là rời nhau. Hơn nữa, K_i là một tập con lớn nhất các giá trị đối với A được chứa trong các điều kiện đơn trên thuộc tính A của cùng một tập câu truy vấn trong H vì K_i được xác định khi có các điều kiện đơn trên thuộc tính A của một tập câu truy vấn trong H chỉ chứa chính xác tập các giá trị trong K_i và không thể tách nhỏ hơn nữa K_i ứng với một tập H xác định.

Dễ thấy, độ phức tạp của Thuật toán 1 là $O(ml)$ với l là lực lượng của K và m là lực lượng của $Dom(A)$.

Thuật toán 2. Phân cụm dữ liệu

Vào: Quan hệ R.

- Tập H các truy vấn trên R. Với mỗi $Q \in H,\ dk(Q)$- ký hiệu điều kiện của truy vấn $Q,\ p(Q)$ ký hiệu xác suất của truy vấn Q.

Ra: - Tập các cụm sở thích C_j và $p(C_j)$ tương ứng với $j = 1, k$.

Cách tính:

1. Đặt $A = \{A_i / A_i$ là thuộc tính của R có xuất hiện trong $H,\ i = 1, m\}$.

2. Với mỗi thuộc tính định lượng A_i trong A, xác định $I(A_i)$.

3. Với mỗi thuộc tính định tính A_i trong A, xác định $K(A_i)$ sử dụng Thuật toán 1.

4. Với mỗi A_i trong A, đặt $P(A_i) = I(A_i)$ hoặc $P(A_i) = K(A_i)$ tùy thuộc vào A_i là thuộc tính định lượng hay định tính. Giả sử $P(A_i) = \{H_{ij} / j = 1, h_i\}$.

5. Xác định tập tất cả các điều kiện cụm với một điều kiện cụm có dạng $\wedge_i dk(A_i)$ với $dk(A_i) = (A_i \in H_{ij}),\ j \in 1,\ h_i, A_i \in \boldsymbol{A}$. Giả sử chúng ta xác định được tập các điều kiện cụm $\boldsymbol{Q} = \{q_j / j = 1, l\}$.

6. Tính $C = \{C_j / C_j$ gồm các bộ của R thoả điều kiện cụm q_j và $C_j \neq \phi,\ j = 1, l\}$.

Giả sử $\boldsymbol{C} = \{C_j / j = 1, k\}$.

7. Tính $p(C_j) = \sigma_i p(Q_i)$ với các Q_i thoả $dk(C_j) \subset dk(Q_i)$, ở đây $dk(C_j)$ kí hiệu điều kiện của cụm $C_j,\ j = 1, k$.

Mệnh đề 2. *Thuật toán 2 xác định đúng tập các cụm sở thích cùng với xác suất cụm tương ứng với một tập H cho trước.*

Chứng minh. Do H_{ij} với $i = 1, m,\ j = 1, h_i$ là một khoảng ưa thích đối với thuộc tính định lượng, hoặc là một tập giá trị ưa thích đối với thuộc tính định tính, có nghĩa là một tập con hay một khoảng con lớn nhất các giá trị đối với A được chứa trong các điều kiện đơn trên thuộc tính A của cùng một tập câu truy vấn trong H nên một cụm C_i với điều kiện $dk(C_i)$ được xác định theo Thuật toán 2 sẽ là một tập con lớn nhất các bộ của R thuộc các kết quả của cùng một tập câu truy vấn trong H. Hơn nữa, đối với một thuộc tính $A_i \in \boldsymbol{A}$, do các H_{ij} với $j = 1, h_i$ là rời nhau theo Thuật toán 1, tập các cụm sở thích xác định được theo Thuật toán 2 là rời nhau và tạo thành một phân hoạch đối với quan hệ R.

Có thể thấy, số các cụm sở thích tối đa là $\Pi_{i=1}^{m} h_i$. Tuy nhiên, do tập cụm sở thích là một phân hoạch đối với R nên $k = n$ với n là số bộ của R.

Nếu các thuộc tính của R xuất hiện trong H khá tập trung hay m không lớn và số các khoảng ưa thích hay các tập giá trị ưa thích đối với các thuộc tính này sau khi đã cắt tỉa và trộn các truy vấn trong tập H cũng không lớn, số các cụm sở thích là không quá lớn hay có thể chấp nhận được. Hơn nữa, giải thuật này tiến hành tự động off-line ở phía hệ thống nên cũng không cần quan tâm nhiều đến độ phức tạp của giải thuật. Điều đáng nói là, thông qua Thuật toán 2, đã đưa ra được các mô tả điều kiện cụm cho các cụm sở thích. Điều này là rất có ý nghĩa trong việc sinh cây điều hướng.

4. SINH CÂY ĐIỀU HƯỚNG

Việc lưu trữ các thông tin về các sở thích này trong CSDL bằng cách thêm vào các nhãn cho các bộ, thông báo bộ đó thuộc cụm sở thích nào.

Khi có một truy vấn mới, ta sử dụng nhãn của các bộ trong kết quả truy vấn làm cơ sở sinh cây điều hướng. Một trong những mục tiêu của cây điều hướng là phân tách được các kết quả đáng quan tâm với các kết quả không đáng quan tâm, trong đó mức độ quan tâm của người dùng tới mỗi nhãn C_j được biết là xác suất truy vấn cụm C_j.

4.1. Chọn thuộc tính chia

Như đã trình bày trong Mục 2, mô hình cây điều hướng tương tự như mô hình cây quyết định. Quá trình xây dựng cây điều hướng do đó cũng đòi hỏi tìm ra thuộc tính chia tốt nhất để mở rộng một nút trên cây. Sử dụng các mô tả điều kiện của các cụm sở thích từ Thuật toán 2, việc chia nhánh tại một nút con theo thuộc tính chia A_i được thực hiện với $P(A_i) = \boldsymbol{I}(A_i)$ hoặc $P(A_i) = \boldsymbol{K}(A_i)$ tùy thuộc vào A_i là thuộc tính định lượng hay định tính. Có hai tiêu chí quan trọng để lựa chọn thuộc tính chia:

Tiêu chí 1. *Cây điều hướng phải phân chia tốt nhất các kết quả theo mức độ quan tâm của người dùng.*

Trong khai phá dữ liệu, Entropy được sử dụng vào bài toán học cây quyết định như một độ đo về tính nhiễu loạn thông tin của một tập dữ liệu luyện bất kỳ [5]. Ở đây, sử dụng ý tưởng Entropy để mô tả mức độ hỗn tạp của các cụm trong một kết quả truy vấn. Ta có định nghĩa sau.

Định nghĩa 4. Cho A_i là một thuộc tính trên quan hệ R, X là một tập kết quả trên R. Hàm *Gain* của X theo A_i, kí hiệu $Gain(X, A_i)$, thể hiện độ giảm entropy trung bình khi phân chia X thành các tập con dựa trên A_i, và được tính theo công thức sau:

$$Gain(X, A_i) = (E(X) - \sum_{H_{ij} \in P(A_i)} (|X_{H_{ij}}|/|X|)E(X_{H_{ij}}),$$

trong đó $E(X)$ kí hiệu Entropy của tập X theo các cụm sở thích, $P(A_i) = \{H_{ij}/j = 1, h_i\}$ là tập các khoảng ưa thích hay tập các giá trị ưa thích của A_i trên tập X, $X_{H_{ij}}$ là tập con của X thoả điều kiện $A_i \in H_{ij}$.

Theo Định nghĩa 4, *Gain* càng lớn khi h_i của A_i càng lớn. Để khắc phục điều này, theo [5], thay thế *Gain* bởi $IGainRatio(X, A_i) = Gain(X, A_i)/E(X, A_i)$, trong đó $E(X, A_i)$ là mức độ nhiễu loạn thông tin của X theo các H_{ij} với $j = 1$, h_i của A_i.

Tiêu chí 2. *Chi phí duyệt cây điều hướng để tìm tất cả thông tin cần quan tâm phải là tối thiểu.*

Với mỗi nút v trên cây điều hướng, sẽ phải tính chi phí duyệt tập dữ liệu (duyệt bộ) của v. Một mặt, người dùng phải duyệt hết $|N(v)|$ kết quả trong v. Mặt khác, để duyệt bộ v, trước đó người dùng phải chọn duyệt nút v thay vì bỏ qua. Gọi $P(v)$ là xác suất người dùng chọn duyệt v, ta có chi phí duyệt bộ nút v là

$$TCost(v) = P(v) \times |N(v)| \text{ với } P(v) = \sigma_{C_j \cap N(v) \neq \phi} p(C_j).$$

Chi phí duyệt tập $Chil(v)$ (gồm các nút con của v) được tính bằng tổng các chi phí duyệt bộ trên từng nút con. Giá trị này càng nhỏ hơn $TCost(v)$ bao nhiêu thì phép phân chia v thành $Chil(v)$ càng hiệu quả bấy nhiêu. Ta có định nghĩa sau.

Định nghĩa 5. [2] Cho A là một thuộc tính trên quan hệ R, v là một nút trên cây điều hướng T ứng với một kết quả truy vấn trên R. Tỷ số gia giảm chi phí duyệt v theo A, ký hiệu $TGainRatio(v, A)$, là đại lượng đặc trưng cho độ giảm chi phí duyệt bộ cây điều hướng tại nút v, khi phân chia v thành các nút con dựa theo các giá trị của thuộc tính A. $TGainRatio(v, A)$ được tính bởi công thức sau

$$TGainRatio(v, A) = \sum_{v_i \in Chil(v,A)} TCost(v_i)/TCost(v),$$

trong đó $Chil(v, A)$ là tập các nút con của v trên T khi phân chia $N(v)$ theo A.

4.2. Kết hợp các độ đo

Vấn đề còn lại là làm thế nào kết hợp hai tỷ số trên thành một độ đo *Gain* duy nhất. Dễ thấy *Gain* tỷ lệ thuận với *IGainRatio* và tỷ lệ nghịch với *TGainRatio*. Vì thế, để đơn giản quá trình tính toán, ta sử dụng tỷ số giữa *IGainRatio* và *TGainRatio* làm công thức tính độ đo *Gain*.

Định nghĩa 6. Cho A là một thuộc tính trên quan hệ R, v là một nút trên cây điều hướng T ứng với một kết quả truy vấn trên R. Tỷ số gia tăng của v theo A, ký hiệu $Gain(v, A)$, là đại lượng đặc trưng cho độ tốt của A khi được chọn làm thuộc tính chia của v, và được tính bởi công thức sau

$$Gain(v, A) = IGainRatio(N(v), A)/TGainRatio(v, A).$$

4.3. Giải thuật sinh cây điều hướng

Cuối cùng là xây dựng giải thuật sinh cây điều hướng cho một kết quả truy vấn. Vì bài toán tìm cây quyết định tối ưu đã được chứng minh là bài toán $\mathcal{NP}$ -đầy đủ, bài toán tìm cây điều hướng tối ưu cũng không có một giải thuật đủ tốt trong mọi trường hợp. Do vậy, ta sẽ tiếp cận lời giải theo hướng các thuật toán xấp xỉ.

Bắt đầu từ gốc, dựng dần các nút con và các nhánh cây con tại các nút.

Tại một nút v, nếu các bộ trong $N(v)$ đều thuộc cùng một cụm, ta dừng lại và v trở thành một nút lá. Trong trường hợp ngược lại, ta chọn thuộc tính A cho giá trị $Gain(v, A)$ lớn nhất làm thuộc tính chia của v. Nếu A_i được chọn làm thuộc tính chia, tiến hành mở rộng v với các nút con có điều kiện chọn dạng $A_i \in H_{ij},\ j = 1, h_i$ và $A_i \in p^-(A_i)$.

Sau đây là thuật toán hình thức của giải thuật sinh cây điều hướng. Thuật toán sử dụng ý tưởng của giải thuật C4.5 cải tiến [5].

Thuật toán 3. Sinh cây điều hướng-$SinhCayDH(r, \boldsymbol{A}, D_Q, \boldsymbol{P(C)}, \theta)$.
Đầu vào: - Tập $D_Q = \{r_1, r_2, ..., r_m\}$ chứa kết quả truy vấn Q.

- Tập A lưu các thuộc tính xuất hiện trong các truy vấn thuộc H.

$P(A_i) = \{H_{ij}/j = 1, h_i\}$, với $A_i \in \boldsymbol{A}$.

- Tập $P(C)$ lưu xác suất truy vấn cụm.

- Ngưỡng dừng θ.

- Gốc ∇ với $N(r) = D_Q$.

Đầu ra : Cây điều hướng T ứng với gốc r.

Ghi chú: // Điều kiện chọn của nút v ký hiệu là $dk(v)$, $dk(r) = dk(Q)$.

// Nhãn cụm của một bộ r_i trong D_Q kí hiệu là $C(r_i)$.

1. if $\forall r_i \in D_Q,\ C(r_i)$ bằng nhau then return.
2. for each $A_i \in \boldsymbol{A}$, tính $Gain(v, A_i)$.
3. Chọn A_i với $Gain(v, A_i)$đạt max
4. if $Gain(v, A_i) < \theta$ then return
5. $A = A \backslash \{A_i\}$.
6. Tạo các nút $\{v : dk(v) = (A_i \in H_{ij}),\ H_{ij} \in P(A_i)$ hoặc $dk(v) = (A_i \in p^-(A_i))\}$ và các nhánh (r, v). Kí hiệu V là tập các nút đã được tạo có kết quả khác rỗng.
7. for each $v_i \in V$: $SinhCayDH(v_i, \boldsymbol{A}, N(v_i), \boldsymbol{P(C)}, \theta)$.
8. end.

Theo Thuật toán 2, với mỗi cụm C_j có $dk(C_j) = \wedge_i dk(A_i)$ với $dk(A_i) = (A_i \in H_{ij}),\ j \in 1,\ h_i, A_i \in \boldsymbol{A},\ i = 1, m$ với xác suất cụm $p(C_j)$. Giả sử trong D_Q có m_j bộ nhãn C_j với $j = 1, k$. Có thể mã hoá các thông tin này trong một bảng $Temp$ với k dòng ứng với k cụm sở thích, m cột đầu tiên ứng với m điều kiện đơn của cụm trên m thuộc tính của $\boldsymbol{A}$, cột $m+1$ ứng với xác suất cụm, cột $m+2$ ứng với số bộ trong D_Q được gán nhãn cụm tương ứng, khi đó $Gain$ được tính rất dễ dàng và độ phức tạp của bước 2 chỉ là $O(mk)$. Do vậy, trong trường hợp tồi nhất, người dùng duyệt cây đến tận nút lá cho tới lần chia nhóm cuối cùng thì độ phức tạp của thuật toán cũng chỉ là $O(m^2k)$.

Nhận xét. Trong [2], Z.Chen trình bày thuật toán BuildTree sinh cây điều hướng dựa trên hàm Gain. Đối với một thuộc tính định tính được chọn làm thuộc tính chia, BuildTree phải tạo một nhánh đối với mỗi giá trị có thể trong miền giá trị của thuộc tính này. Trong trường hợp thuộc tính này có quá nhiều giá trị xuất hiện trong CSDL, nó sẽ sinh ra quá nhiều nhánh mà cách giải quyết trong [2] là không rõ ràng và đòi hỏi thêm các tri thức từ bên ngoài, chẳng hạn các phân cấp khái niệm hay các tri thức phụ thuộc miền được đưa vào bởi các chuyên gia. Đối với một thuộc tính định lượng A_i được chọn làm thuộc tính chia, với mỗi giá trị v của A_i được chọn, BuildTree tạo hai nhánh với điều kiện $A_i = v$ và $A_i > v$. Quá trình chia hai nhánh này phải xem xét mọi vị trí chia có thể (giá trị số xuất hiện trong CSDL) để chọn một vị trí sinh ra phân hoạch tốt nhất có thể. Z.Chen cũng đề cập đến khả năng chia nhiều nhánh thông qua việc xem xét mọi vùng chia có thể của các giá trị số có thể đối với thuộc tính số, tuy nhiên, điều này là không khả thi vì số vùng có thể là rất lớn và tác giả đã sử dụng chia hai nhánh trong BuildTree. Độ phức tạp của BuildTree trong [2] là $O(mnklogn)$, với m là số thuộc tính, n là số bộ (bản ghi) và k là số cụm sở thích. Thuật toán 3 đã khai thác ý tưởng sinh cây điều hướng BuildTree của Z.Chen, tuy nhiên, cách tính $Gain(X, A_i)$ và $IGainRatio(X, A_i)$ ở đây hiệu quả hơn do sử dụng các $P(A_i)$. Hơn nữa, với thuộc tính chia được chọn là A_i, Thuật toán 3 chỉ cần tạo hi nhánh tương ứng với $P(A_i) = \{H_{ij}/j = 1, h_i\}$. Theo đánh giá, độ phức tạp của Thuật toán 3 là $O(m^2k)$ với m là số thuộc tính và k là số cụm sở thích. Thực tế, số các thuộc tính xuất hiện trong các câu truy vấn trong tập truy vấn quá khứ là khá nhỏ so với số bộ của một quan hệ hay một khung nhìn được tra cứu, vì vậy mà $m << n$. Hơn nữa, theo Thuật toán 2, do tập cụm sở thích là một phân hoạch đối với R nên $k = n$ với n là số bộ của R. Ngoài ra, đã sử dụng một số heuristic tiền xử lý tập các truy vấn quá khứ đã trình bày trong mục 3 của bài báo để giảm k và để thu được các cụm sở thích có ý nghĩa. Với các nội dung trên, bài báo đã đề xuất một cách tiếp cận khá đơn giản, thuật toán được cài đặt khá hiệu quả cho phép sinh cây điều hướng trợ giúp cho những người dùng mới truy cập và khai thác CSDL, nhanh chóng tìm được các kết quả mong muốn.

5. KẾT LUẬN

Giải thuật phân cụm dữ liệu được thực hiện dựa trên việc phân tích các câu hỏi đã đặt ra trong quá khứ của tất cả người dùng để phát hiện tri thức về sở thích, về mức độ quan

tâm của người dùng đối với các giá trị dữ liệu trong hệ thống. Tri thức này được tích hợp vào các nhãn trong mỗi dữ liệu. Giải thuật này tiến hành tự động off-line ở phía hệ thống không cần quá trình tương tác từ phía người dùng.

Giải thuật sinh cây điều hướng được thực hiện mỗi khi có kết quả trả về với số lượng lớn. Giải thuật này phân tích các nhãn trong từng dữ liệu của tập kết quả, sử dụng các độ đo để tìm ra thuộc tính chia nhỏ nhất. Giải thuật này sử dụng ý tưởng sinh cây quyết định C4.5 của Quinlan, với các cải tiến để việc lựa chọn thuộc tính chia chính xác hơn trong mô hình đặc thù của cây điều hướng.

Ý tưởng "phân loại dữ liệu" đã được đề cập đến từ lâu trong nhiều ngành khác nhau của trí tuệ nhân tạo. Trong lĩnh vực khai phá dữ liệu và xử lý tri thức (KDD), đã có nhiều nghiên cứu về vấn đề phân lớp (classification), phân cụm (clustering) trên các kho dữ liệu (data warehouse), phục vụ công tác phân tích, phát hiện mẫu, dự đoán xu hướng, hỗ trợ ra quyết định,.. Ngoài ra cũng có nhiều nghiên cứu về phân loại văn bản (text categorization), phân cụm kết quả tìm kiếm trên web (web search result clustering). Tuy nhiên, bài toán phân loại kết quả truy vấn trên CSDL quan hệ có những đặc thù và khó khăn riêng so với các tiếp cận trên.

Thứ nhất, phân loại kết quả truy vấn trên CSDL quan hệ đòi hỏi khả năng xử lý trên nhiều kiểu dữ liệu khác nhau (bao gồm các dữ liệu định tính categorical data và các dữ liệu định lượng numerical data), trong khi phân loại văn bản và kết quả tìm kiếm trên web thường chỉ xử lý trên đối tượng là từ vựng, hoặc siêu dữ liệu (metadata). Cả hai chỉ là các dạng khác nhau của dữ liệu định tính.

Thứ hai, các giải thuật phân cụm và phân lớp thường được tiến hành off-line, trên không gian thuộc tính đã biết trước. Trong khi đó, nghiên cứu này đề cập đến việc phân loại on-line (tự động mỗi khi có kết quả trả về với số lượng lớn) trên không gian thuộc tính chưa biết trước. Vì thế, bên cạnh tính chính xác, việc phân loại còn đòi hỏi cao ở yếu tố hiệu năng (khả năng phản hồi cho người dùng trong thời gian sớm nhất).

Thứ ba, mục tiêu của các quá trình là khác nhau. Mục tiêu của phân cụm là để giảm thiểu độ tương tự (similarity) giữa các cụm và phát hiện các mẫu bị che (hidden patterns), phục vụ cho học máy (machine learning). Mục tiêu của phân loại kết quả truy vấn là giảm thời gian và công sức của người dùng trong việc lọc bỏ các thông tin thừa mà họ không quan tâm. Do đó, quá trình phân loại phải mang tính hướng người dùng cao.

Đối với vấn đề phân loại kết quả truy vấn SQL, hiện nay mới có hai công trình đáng chú ý [1,2]. Cả hai công trình này đều hướng đến giải pháp phân cấp kết quả truy vấn SQL theo cấu trúc cây, giúp người dùng nhanh chóng định vị các nhóm kết quả phù hợp nhất với mong muốn của mình. Trong mô hình cây điều hướng, việc quyết định khi nào dừng phân chia kết quả không đơn giản. Chakrabarti [1] lựa chọn dừng phân chia khi số kết quả đủ nhỏ. Z. Chen [2] cố gắng xây dựng các lớp dữ liệu cho kết quả, để mô phỏng lại các điều kiện dừng cây quyết định. Trong bài báo này, giải pháp được đưa ra là để người dùng tự lựa chọn khi nào thì dừng duyệt cây. Thuật toán sinh cây điều hướng thực hiện các tính toán khá đơn

giản và không cần nhiều bộ nhớ lưu trữ các kết quả tính toán với độ phức tạp là chấp nhận được. Giải pháp này có thể được cài đặt để thực hiện phân loại các kết quả truy vấn đối với các CSDL có kích thước khá lớn.

TÀI LIỆU THAM KHẢO

[1] K. Chakrabarti, S. Chaudhuri, S. won Hwang, Automatic categorization of query results, *Proceedings of the 23rd ACM SIGMOD Conference*, Paris, France, 2004 (755–766).

[2] Z. Chen, T. Li, Adressing diverse user preferences in SQL-Query-Result navigation, *Proceedings of the 26th ACM SIGMOD Conference*, Beijing, China, 2007 (641–652).

[3] M. Merzbacher, W. W. Chu, Pattern-based clustering for database attribute values, *Proceedings of AAAI Workshop on Knowledge Discovery in Databases*, Washington, DC, USA, 1993 (291–298).

[4] W. Kieling, Foundations of preferences in database systems, *Proceedings of the 28th VLDB Conference,* Hong Kong, China, 2002 (311–322).

[5] J. R. Quinlan, *C4.5: Programs for machine learning*, Morgan Kaufmann Publishers, San Francisco, CA, USA, 1993.

Nhận bài ngày 17 - 7 - 2008
Nhận lại sau sửa ngày 20 - 8 -2009