TRÍ TUỆ NHÂN TẠO

**ỨNG DỤNG THUẬT TOÁN CÂY QUYẾT ĐỊNH**

**HỖ TRỢ NHÂN VIÊN GỬI THƯ**

**TƯ VẤN KHÁCH HÀNG**

Giáo viên hướng dẫn : TS. Nguyễn Nhật Quang

Sinh viên thực hiện : Vũ Thành Trung – 20073070

Nguyễn Hồng Phúc – 20072236

Lưu Văn Đảng –

Nguyễn Văn Hưng –

Lớp : Truyền thông & Mạng

Khóa : 52

**LỜI MỞ ĐẦU**

Hiện nay, trong rất nhiều các lĩnh vực, ngành nghề hay những công việc cụ thể, vai trò của con người vẫn luôn là trung tâm. Tuy nhiên, thực tế cho thấy, ngày càng có nhiều sự hỗ trợ đắc lực từ phía máy móc và công nghệ trong những công việc đó. Thật vậy, có thể nói vai trò của khoa học và công nghệ cao là không thể thiếu và đang ảnh hưởng rất lớn tới đời sống của chúng ta. Đó là những công cụ hết sức đắc lực của con người, giúp con người giải quyết vấn đề nhanh chóng, chính xác và hiệu quả hơn, tiết kiệm rất nhiều chi phí.

Một ứng dụng dễ nhận thấy nhất về sự trợ giúp của khoa học công nghệ đối với đời sống con người đó là: hỗ trợ con người ra quyết định. Đó là việc áp dụng trí tuệ nhân tạo vào những công việc tưởng chừng như quá khó, mất rất nhiều công sức và thời gian. Qua quá trình tìm hiểu và khảo sát, nhóm chúng em đã quyết định chọn đề tài Ứng dụng trí tuệ nhân tạo trong hệ thống CRM và cụ thể là Cây quyết định hỗ trợ nhân viên chăm sóc khách hàng gửi thư tư vấn khách hàng. Nếu trước đây, những nhân viên phải thu thập thông tin từ hàng trăm, hàng ngàn khách hàng, đối sánh với thông tin các lớp, các khóa học phù hợp rồi gửi thư mời, tư vấn cho khách hàng tham gia học tập. Đó là một vấn đề hết sức nan giải và đòi hỏi chi phí công sức rất lớn. Tuy nhiên, với hiệu quả rất cao của các thuật toán Cây quyết định, những công việc đó sẽ được giảm tải chi phí xuống mức thấp nhất có thể, hỗ trợ con người rất tốt.

Trong phạm vi đề tài cũng như kiến thức, nhóm chúng em đã cố gắng đưa ra được một bài toán rất nhỏ nằm trong một mảng đề tài rất lớn. Với sự hướng dẫn, giúp đỡ rất tận tình của thầy Nguyễn Nhật Quang, các thành viên trong nhóm đã hiểu được ý nghĩa thuật toán, gắn được vào bài toán thực tế ở mức có thể. Chúng em cũng xin cảm ơn sự đóng góp, hỗ trợ của thầy giáo và mong muốn có thể triển khai tiếp đề tài này ở mức rộng hơn, hoàn thiện hơn trong tương lai. Chúng em xin chân thành cảm ơn!

Nhóm thực hiện

**NỘI DUNG**

1. Giới thiệu đề tài:
   1. Đặt vấn đề:

Ở mỗi công ty, trung tâm quản lí đào tạo, khách hàng sẽ đến và tìm hiểu những dịch vụ cung cấp đào tạo, tìm ra cho mình những khóa học, lớp học và các yếu tố phù hợp để có thể lựa chọn một hình thức đào tạo tốt nhất. Công việc này sẽ được những nhân viên chăm sóc khách hàng cung cấp tới khách hàng những thông tin mong muốn, với những tư vấn chi tiết, hợp lí nhất.

Tuy nhiên, với một lượng dữ liệu thông tin khách hàng rất lớn, công việc lựa chọn những khóa học, lớp học phù hợp với từng đối tượng khách hàng là việc hết sức khó khăn, đòi hỏi tiêu tốn rất nhiều chi phí thời gian và công sức.

Việc áp dụng thuật toán Cây quyết định vào xử lí dữ liệu thông tin đó là giải pháp rất tốt. Với những tiêu chí của từng khóa học tương ứng với những nhu cầu của khách hàng, nhân viên chỉ cần dùng giao diện của chương trình để thao tác, chương trình sẽ tự động kiểm tra, so sánh và đưa ra những gợi ý chính xác, phù hợp nhất. Từ đó, nhân viên sẽ biết được đối tượng khách hàng cụ thể đó sẽ phù hợp với những khóa đào tạo nào. Cuối cùng, nhân viên sẽ gửi thư tư vấn đến khách hàng, cung cấp cho khách hàng những thông tin hữu ích nhất.

* 1. Mô tả bài toán:
* Đầu vào: tập dữ liệu thông tin đầu vào sẽ bao gồm:
* Thông tin khóa học/lớp học: sẽ gồm các thông tin như nội dung môn học, chứng chỉ sau khi học, giảng viên, ca học, yêu cầu kiến thức...
* Thông tin khách hàng: trình độ học vấn, mục đích học tập, mong muốn chứng chỉ sau khi học, ca học phù hợp, yêu cầu giảng viên...
* Đây là những thông tin sẽ được lưu trong Cơ sở dữ liệu của trung tâm và sẽ được xử lí ở bước tiếp theo
* Xử lí: với các dữ liệu đầu vào như trên (dataset), chương trình sử dụng thuật toán Cây quyết định (cụ thể là ID3) để xử lí dữ liệu, đưa ra mô hình cây quyết định tương ứng với các dữ liệu đã có. Từ đó trả ra giá trị đánh giá phù hợp nhất giữa thông tin của khách hàng và thông tin các khóa học: khách hàng nào phù hợp với lớp học nào...
* Đầu ra: dữ liệu trả ra sẽ là kết quả tương ứng khách hàng – lớp học. Nhân viên tư vấn sẽ dựa vào những kết quả đó và gửi thư tư vấn đến khách hàng, cung cấp những thông tin chi tiết về lớp học phù hợp nhất với khách hàng, hỗ trợ khách hàng có thể lựa chọn cho mình một lớp học nào đó thích hợp nhất.

Mô hình làm việc của bài toán

**OUTPUT**

Kết quả trả về các cặp

*Khách hàng – Khóa học*

**Process**

(Xử lí với tập dữ liệu đầu vào)

**INPUT**

Tập dữ liệu đầu vào (DataSet)

1. Triển khai bài toán:
   1. Thuật toán Cây quyết định:
      1. Mô tả thuật toán:

Có khá nhiều thuật toán Datamining hay và được ứng dụng nhiều, nhưng trong phạm vi kiến thức và thời gian nghiên cứu, tìm hiểu thì nhóm chúng em quyết định chọn thuật toán ID3. Một số thuật toán sử dụng trong Datamining:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Thuật toán  Tham số | CART | ID3 và C4.5 | |  | | --- | | SLIQ & SPRINT | | |  | | --- | | Proposed approach | |
| Tính toán | |  | | --- | | Gini Diversity Index | | |  | | --- | | Entropy Info-gain | | |  | | --- | | Gini Index | | |  | | --- | | Info gain & Uncertainity coefficient | |
| Phương thức | |  | | --- | | Constructs Binary Decision Tree | | |  | | --- | | Top-Down Decision Tree Construction | | |  | | --- | | Decision Tree Construction in a Breadth first manner | | |  | | --- | | Decision Tree with concepts of node merging and Height –balance using AVL trees | |
| Cắt tỉa | |  | | --- | | Post prun- ing based on cost-complexity measure | | |  | | --- | | Pre-pruning using a single pass algorithm | | |  | | --- | | Post pruning based on MDL principle | | |  | | --- | | Dynamic pruning based on thresholds | |

* Lí do lựa chọn:

+ Cây quyết định dễ hiểu

+ Việc chuẩn bị dữ liệu cho một cây quyết định là cơ bản hoặc không cần thiết

+ Cây quyết định có thể xử lý cả dữ liệu có giá trị bằng số và dữ liệu có giá trị là tên thể loại

+ Cây quyết định là một mô hình hộp trắng

+ Có thể thẩm định mô hình bằng các kiểm tra thống kê

+ Cây quyết định có thể xử lí một lượng lớn dữ liệu trong một khoảng thời gian ngắn

Thuật toán ID3(Iterative Dichotomiser 3) do Quinlan phát triển vào năm 1979, mục đích xây dựng nên một cây quyết định dựa vào tập dữ liệu dataset, với mỗi thuộc tính sẽ có những giá trị đi cùng.

* Mỗi nút (không phải lá) của một cây quyết định tương ứng với một thuộc tính đầu vào, và mỗi một nhánh con đi ra tiếp theo chính là giá trị của thuộc tính đó. Một nút lá tương ứng với giá trị kỳ vọng của các thuộc tính trước đó mà được xác định bằng cách đi từ nút gốc cho đến nút lá đó (ta có thể hiểu đó là kết quả kì vọng cuối cùng được đưa ra sau khi đã duyệt qua tất cả những thuộc tính có liên quan trước đó tuân theo những luật, ràng buộc được nêu ra)
* Một cây quyết định “tốt” là cây có mỗi nút lá tương ứng với một thuộc tính mà thuộc tính đó có giá trị ý nghĩa tốt nhất trong toàn bộ những thuộc tính chưa được duyệt (tính từ nút gốc cho đến nút hiện tại). Tức là, chúng ta muốn dự đoán giá trị của thuộc tính bằng cách dựa vào số lượng những nghi vấn nhỏ nhất trên tổng số những nghi vấn trung bình (mức độ tin cậy và chính xác càng cao thì sẽ được chọn)

1. Entropy:

Entropy được sử dụng để xác định thông tin đầu vào một thuộc tính đặc biệt là về các thuộc tính đầu ra cho một tập hợp các dữ liệu huấn luyện, định mức giá trị cho những nguồn thông tin không chắc chắn, nguồn thông tin đưa vào càng không chắc chắn thì càng cần có thêm thông tin mô tả về nó.

Số lượng trung bình của lượng thông tin cần thiết để xác định mỗi nguồn tin là một thước đo sự không chắc chắn của đối tượng tiếp nhận về nguồn tin đó, và được gọi là entropy của nguồn. Giả thiết rằng nguồn S có *n* thông tin *{m1,m2,..., m­n},* những thông tin này không phụ thuộc vào nhau, ta gọi xác suất của mỗi *mi*là *pi* . Nếu nguồn S tuân theo luật phân bố *P = (p1, p2,..., pn)* thì ta có công thức tính Entropy của *P* là:



1. Info – Gain:

Bây giờ, ta chia thuộc tính T thành những tập dữ liệu con mức dưới T1, T2,..., Tn. Như vậy, lượng thông tin cần để xác định những thành phần trong T chính là lượng thông tin trung bình cần để xác định những đối tượng dữ liệu con của T như trên:



Trong quá trình dựng cây quyết định, ta sẽ luôn cần biết lượng thông tin được cung cấp tương ứng từ mỗi thuộc tính *X*. Đó chính là sự chênh lệch giữa giá trị thông tin cần để phân chia các thành phần của T trước khi biết giá trị của *X, H(T)* và giá trị cần để phân chia thành tập dữ liệu con của *T* sau khi đã biết giá trị của *X, H(T)*. Từ đó ta có khái niệm Information Gain cho thuộc tính *X* cùng với tập dữ liệu *T* của nó:

*Gain (X,T) = H(T) – H(X,T)*

* + 1. Ý tưởng thuật toán :

ID3(D, Target, Atts)

* Kết quả trả về : một cây quyết định được xây dựng theo những giá trị đầu vào
* Các biến :

+ D : tập dữ liệu huấn luyện của đầu vào

+ Target : những thuộc tính mà giá trị được dự đoán bởi cây quyết định

+ Atts : tập hợp các thuộc tính cần kiểm thử trong quá trình xây dựng cây (chưa được xét duyệt)

* Thuật toán (giả ngôn ngữ) :

function ID3 (I, 0, T) {

/\* I is the set of input attributes

\* O is the output attribute

\* T is a set of training data

\*

\* function ID3 returns a decision tree

\*/

if (T is empty) {

return a single node with the value "Failure";

}

if (all records in T have the same value for O) {

return a single node with that value;

}

if (I is empty) {

return a single node with the value of the most frequent value of O in T;

/\* Note: some elements in this node will be incorrectly classified \*/

}

/\* now handle the case where we can’t return a single node \*/

compute the information gain for each attribute in I relative to T;

let X be the attribute with largest Gain(X, T) of the attributes in I;

let {x\_j| j=1,2, .., m} be the values of X;

let {T\_j| j=1,2, .., m} be the subsets of T when T is partitioned

according the value of X;

return a tree with the root node labelled X and

arcs labelled x\_1, x\_2, .., x\_m, where the arcs go to the

trees ID3(I-{X}, O, T\_1), ID3(I-{X}, O, T\_2), .., ID3(I-{X}, O, T\_m);

}

* + 1. Ưu điểm của thuật toán ID3:
* Sử dụng thuật tìm kiếm leo đồi (hill - climbing) dựa trên giá trị Gain để tìm kiếm các thuộc tính trong toàn bộ Cây quyết định
* Đầu ra (Output) chỉ là một giả thuyết đơn (1 kết quả duy nhất)
* Không bao giờ gặp hiện tượng quay lui – tính hội tụ cao
* Sử dụng dữ liệu huấn luyện ở từng bước, trái ngược với những thuật giải phát triển mở rộng cây quyết định (có thể hạn chế được kích thước Cây không quá lớn)
* Sử dụng các thuộc tính tĩnh: hạn chế tối đa lỗi cho những bản ghi dữ liệu riêng lẻ, có thể ảnh hưởng tới toàn bộ dự án
* Kiểm soát được dữ liệu rác, dữ liệu tạp bên ngoài bằng cách giảm bớt yêu cầu tiêu chuẩn cho việc chấp nhận những dữ liệu chưa hoàn chỉnh
  + 1. Xây dựng Cây quyết định :
* Cây được thiết lập từ trên xuống dưới (phương pháp top-down)
* Các mẫu huấn luyện nằm ở gốc của cây
* Chọn một thuộc tính để phân chia thành các nhánh. Thuộc tính được chọn dựa trên độ đo thống kê hoặc độ đo heuristic (chính là các giá trị Entropy, Info-Gain đã tính toán ở trên). Với từng thuộc tính, giá trị Gain nào thấp nhất trong tập các thuộc tính chưa được xét thì sẽ được chọn đưa vào cây ở bước đó. Ta có thể hiểu cách lựa chọn đó chính là nhằm mục đích tạo ra một cây nhỏ nhất có thể, giá trị Gain càng nhỏ có nghĩa là thuộc tính đó có lợi nhất cho quá trình phân lớp.
* Tiếp tục lặp lại việc xây dựng cây quyết định cho các nhánh.
* Điều kiện dừng :

+ Tất cả các mẫu rơi vào một nút thuộc về cùng một lớp (nút lá)

+ Không còn thuộc tính nào có thể dùng để phân chia mẫu nữa

+ Không còn lại mẫu nào tại nút

* 1. DataSet :

*(thêm Dataset và tính toán: ông thêm dataset còn tôi tính toán)*

* 1. Cơ sở dữ liệu :

*(Ông cập nhật DB cuối cùng rồi chụp lại cái ảnh Diagram của DB nhá)*

* 1. Công nghệ sử dụng :
* Môi trường lập trình : Microsoft Visual Studio 2008
* Ngôn ngữ lập trình : C# trên nền công nghệ .NET 3.5
* Hệ quản trị cơ sở dữ liệu : Microsoft SQL Server 2005
  1. Mô phỏng chương trình :

*(Ông chụp ảnh màn hình demo cái Cây quyết định và form nhập liệu, form chính của app nhá)*

1. Định hướng phát triển :
   1. Những khuyết điểm của thuật toán ID3 :

* Chỉ thích hợp với mô hình có lượng dữ liệu ít, rời rạc
* Không thích ứng được với những tập dữ liệu tạp (dễ phát sinh lỗi)
* Không hiệu quả khi xuất hiện những dữ liệu không mong muốn
* Cây quyết định khi dựng ra vẫn còn có thể lớn, rườm rà, chưa được tối ưu ở mức tối đa có thể
  1. Cải tiến thuật toán : sẽ sử dụng thuật toán C4.5

1. Thuật toán :

* Mạnh mẽ khi gặp những dữ liệu tạp, có khả năng phòng tránh hiện tượng Overfitting : là hiện tượng lượng dữ liệu không cần thiết (hoàn toàn có thể loại bỏ) vẫn được đưa vào Cây khiến cho kết quả trả về không tối ưu, cây lớn và rườm rà. Ở các thuật toán tiên tiến hơn, vấn đề này đã được giải quyết, kết quả trả về cuối cùng sẽ được tối ưu hóa hơn.
* Thích hợp được với các dữ liệu liên tục
* Giải quyết bài toán với trường hợp mà các thuộc tính có dữ liệu trống (khuyết dữ liệu trong quá trình dựng cây quyết định)
* Có thể chuyển đổi từ Cây quyết định thành các Luật

1. Các phương pháp cắt tỉa cây :

* Pre – pruning : dừng ngay quá trình phát triển nhánh của cây khi gặp dữ liệu không chắc chắn (hoặc dữ liệu tiếp theo là rỗng)
* Post – pruning : tạo ra một cây quyết định hoàn hảo rồi sau đó sẽ cắt bỏ dần những phần dữ liệu không chắc chắn. Ý tưởng này có 2 bước thực hiện :

+ Subtree replacement : loại bỏ đi những đoạn dữ liệu không chắc chắn

+ Subtree rasing : tiếp tục phát triển dữ liệu từ đoạn dữ liệu vừa được cắt bỏ

Quá trình này được lặp đi lặp lại cho đến khi duyệt hết toàn bộ Cây quyết định.

1. Kiểm soát dữ liệu trống (Handling missing value):

Mỗi khi gặp dữ liệu trống trong quá trình xây dựng cây, ta sẽ coi đó là dữ liệu chưa xác định (unknown data) và sẽ biểu diễn dữ liệu đó là dữ liệu “?” trên cây, bỏ qua và tiếp tục phát triển cây với những dữ liệu tốt.

1. Xử lí dữ liệu liên tục (đa gia trị với một thuộc tính):

Đó là những trường hợp mà ở mỗi thuộc tính có nhiều giá trị (not unique), giải pháp cần tính ra xác suất của những giá trị nằm trong thuộc tính đó: *SplitInfo = (X,T).* Như vậy, giá trị Gain lúc này sẽ khác so với ban đầu:



1. Những khó khăn trong quá trình thực hiện:

*(chém gió mạnh đê)*

**KẾT LUẬN**

*(phần này tôi sẽ chém gió mấy phút là xong thôi :D)*

**TÀI LIỆU THAM KHẢO**

*(ae cùng xem là cần những gì ở mục này thì thêm vào cho đủ)*