CẢI TIẾN CÁC THUẬT TOÁN PHÂN LỚP DỮ LIỆU FCM, AVQ VÀ ỨNG DỤNG CHO HỆ LUẬT MỜ XẤP XỈ

Lê Ngọc Thạnh
Trường Đại Học Kinh Tế Tp.HCM
(Bài nhận ngày 10 tháng 4 năm 2004, hoàn chỉnh sửa chữa ngày 30 tháng 6 năm2005)

TÓM TẮT: Khám phá tri thức từ dữ liệu là một vấn đề quan trọng của công nghệ tri thức. Trong đó, vai trò của các thuật toán phân lớp dữ liệu góp phần quyết định đến chất lượng của tri thức thu nhận được. Bài báo này trình bày một số cải tiến cho các thuật toán phân lớp dữ liệu FCM và AVQ. Đồng thời áp dụng các thuật toán cải tiến này vào quá trình học cấu trúc của hệ luật mờ SAM xấp xỉ quan hệ tương quan trong dữ liệu học.

Từ khóa: Phân lớp dữ liệu, Thuật toán Fuzzy C-Means, Thuật toán AVQ. Hệ luật mờ, Mô hình hệ mờ cộng chuẩn.

1. Giới thiệu

Phân lớp dữ liệu là khâu quan trọng trong quá trình tạo tri thức từ dữ liệu. Dựa trên kết quả phân lớp dữ liệu, ta có thể xây dựng các kiến trúc tri thức khác nhau (mạng neuron, hệ luật, hệ luật mờ,...) tùy theo yêu cầu của mô hình nhằm biểu diễn các nội dung tri thức ẩn chứa bên trong dữ liệu. Mô hình hệ luật mờ cộng chuẩn (Standard Additive Model – SAM [1]) là kiến trúc bao gồm các luật mờ hoạt động song song và kết hợp theo mô hình cộng-tích (sum-product) cho phép thể hiện quan hệ tương quan giữa hai đại lượng bất kỳ thông qua việc học các mẩu dữ liệu thống kê phản ánh quan hệ giữa hai đại lượng đó.

Việc sử dụng hệ mờ SAM cho mô hình tương quan xuất phát từ những ưu điểm của kiến trúc này. Với hệ thống các luật mờ, hệ SAM trở nên gần gũi với người sử dụng, cho phép họ tham gia điều chỉnh mô hình bất cứ lúc nào. Ngoài ra, hệ SAM có thể dễ dàng được xây dựng dựa trên các phân lớp dữ liệu do các thuật toán phân lớp dữ liệu tạo ra từ dữ liệu học.

Nội dung bài báo bao gồm ba phần chính. Đầu tiên là phần trình bày hai thuật toán phân lớp dữ liệu và một số cải tiến thuật toán. Phần tiếp theo là sự vận dụng thuật toán cải tiến cho hệ mờ SAM. Cuối cùng là ứng dụng kết quả nghiên cứu cho vấn đề xấp xỉ tương quan và dự báo.

2. Thuật toán phân lớp dữ liệu

Phân lớp dữ liệu là quá trình gom các mẩu dữ liệu thành từng lớp. Mỗi lớp là một tập hợp các mẩu dữ liệu tương tự nhau theo một số đặc điểm, tính chất nhất định. Nhiệm vụ của phân lớp dữ liệu là tối đa hóa tính tương tự giữa các mẩu dữ liệu trong một lớp và tối tiểu hóa tính tương tự giữa các lớp với nhau. Đây là tiêu chuẩn đánh giá chất lượng của thuật toán phân lớp dữ liệu, phản ánh mức độ hỗn loạn của dữ liệu sau khi thực hiện phân lớp.

Phát biểu bài toán phân lớp dữ liệu:

Cho D là tập hợp *ntd* mẩu dữ liệu phản ánh quan hệ giữa hai đại lượng X, Y trong không gian tích XY, $ntd \in N$.

$D = \{ xy_j \in XY,\ j = \overline{1, ntd} \}$

Gọi d(a,b) là độ khác biệt giữa a, b ∈ XY. Hãy xác định một phân lớp $\Phi_D(V, U)$ trên D

- V : Tập các vector trọng tâm của các lớp,
  $V = \{v_i \in XY,\ i = \overline{1,c}\}$,
  $c \in N$ được chọn trước: $c \le ntd$
- U : Tập các giá trị chỉ số thuộc các lớp,
  $U = \{u_{ij} \in [0,1],\ i = \overline{1,c},\ j = \overline{1,ntd}\}$

  u_{ij} : Chỉ số thuộc về lớp xác định bởi vector v_i của mẩu dữ liệu xy_j.

Sao cho:

$$\sum_{i=1}^{c} u_{ij} = 1\ ,\ \forall j,\ j = \overline{1,ntd} \qquad (1)$$

$$J(U,V) = \sum_{j=1}^{ntd}\sum_{i=1}^{c} u_{ij}.d(xy_j, v_i) \to \min \qquad (2)$$

Việc giải bài toán phân lớp dữ liệu được đưa về việc giải bài toán tối ưu phát biểu bởi (1) và (2). Nhằm tránh việc giải bài toán trên bằng phương pháp đại số khá phức tạp, máy học đề ra phương pháp lặp cho bài toán như sau:

b1. Khởi tạo ngẫu nhiên $\{u_{ij}\}$ thỏa (1)
b2. Xác định các $\{v_i\}$ theo (1) và (2)
b3. Tính lại các u_{ij} theo $\{v_i\}$ vừa tìm được.
b4. Xét điều kiện chấp nhận được của $\{u_{ij}\}$
b5. Nếu $\{u_{ij}\}$ chưa đạt thì đến b2
b6. Dừng thuật toán

Khi thuật toán kết thúc, V xác định trọng tâm các phân lớp, U xác định sự phân bổ của các mẩu dữ liệu vào các phân lớp nói trên.

Hai thuật toán phân lớp dữ liệu phổ biến sử dụng phương pháp tiếp cận nói trên là Fuzzy C-Means và AVQ. Phần sau đây sẽ trình bày chi tiết và cải tiến của các thuật toán này.

2.1. *Thuật toán FCM (Fuzzy C-Means)* [3]

Do Bezdek công bố 1981. Thuật toán mờ hóa chỉ số thuộc lớp trong hàm mục tiêu thông qua giá trị độ mờ m; $1 \le m \le 2$, và độ khác biệt d được định nghĩa bằng khoảng cách euclic như sau:

$$J(U,V) = \sum_{xy_j \in D}\sum_{i=1}^{c} u_{ij}^{m} \left\| xy_j - v_i \right\|^2 \to \min \qquad (3)$$

$$d(a,b) = \left\| a - b \right\|^2 \qquad (4)$$

Lấy đạo hàm riêng của J(U,V) theo hai biến U và V, phối hợp với phương trình (1), ta được:

Phương pháp lặp cho việc tìm nghiệm U và V của hệ phương trình (5) và (6) được thực hiện theo trình tự sau:

- Tạo ngẫu nhiên U thỏa công thức (1)

$$v_i = \frac{\sum_{j=1}^{ntd} u_{ij}^{m}.xy_j}{\sum_{j=1}^{ntd} u_{ij}^{m}} \qquad (5) \qquad u_{ij}^{m} = \frac{\left(\frac{1}{\left\| xy_j - v_i \right\|^2}\right)^{\frac{1}{1-m}}}{\sum_{k=1}^{c}\left(\frac{1}{\left\| xy_j - v_k \right\|^2}\right)^{\frac{1}{1-m}}} \qquad (6)$$

- Xác định V theo công thức (5)
- Tính lại U theo V nhờ công thức (6)

- Lặp lại việc điều chỉnh V và U đến khi U nhận giá trị hội tụ của nó tại lần lặp thứ t_n : với $\varepsilon > 0$ cho trước, $\forall t > t_n$. $e = \max_{ij}\left(\left\|u_{ij}(t+1) - u_{ij}(t)\right\|\right) < \varepsilon$ (7)

Chi tiết thuật toán Fuzzy C-Means

b1. Khởi tạo ngẫu nhiên $\{u_{ij}\}$ thỏa (1)

b2. Xác định các v_i theo (5), $i = \overline{1,c}$

b3. Tính lại các u_{ij} dựa trên V mới theo (6)

b4. Kiểm tra mức chấp nhận của U theo (7)

b5. Nếu U chưa chấp nhận được thì đến b2

b6. Dừng thuật toán

Cải tiến thuật toán

Trong các thuật toán phân lớp dữ liệu, số phân lớp c được giả định là hằng số. Tuy nhiên giá trị này không phải luôn giống nhau trên các bộ dữ liệu khác nhau. Vấn đề là làm cách nào để thuật toán phân lớp dữ liệu tự động xác định số phân lớp phù hợp trên mỗi bộ dữ liệu xử lý.

Giải pháp heuristic do [3] đề ra khá phức tạp và không có cơ sở lý luận để kiểm chứng. Parag [4] với thuật toán *Fuzzy-Ants,* Carla [5] với thuật toán *FCV-TSD* cung cấp giải pháp phân bổ hợp lý các phân lớp để quá trình hậu xử lý với thuật toán FCM đạt hiệu quả cao. Tuy nhiên, qui trình xử lý của [4] có tính ngẫu nhiên, [4] và [5] đều không giải quyết vấn đề xác định số phân lớp C. Bagirov [6] với thuật toán *Optimization Clustering* là một tiếp cận hoàn chỉnh cho vấn đề này nhưng hàm đánh giá V của [6] quá phức tạp; mỗi phân lớp của V được xét lại trên toàn bộ D với chi phí tính toán rất lớn. Thuật toán *Greedy K-Means* của N.Hussein [7] tốt hơn [6] vì chỉ cập nhật V mỗi khi lấy một mẩu dữ liệu từ D, nhưng việc này vẫn đòi hỏi một khối lượng tính toán khá lớn. Thuật toán cải tiến mà bài báo này trình bày xuất phát từ ý tưởng : "*Quá trình phân lớp dữ liệu phát triển theo số mẩu dữ liệu tiếp nhận. Phân lớp chỉ phát sinh khi có thêm đặc tính mới trên mẩu dữ liệu mới*".

Ý tưởng này được áp dụng cho việc xác định số phân lớp c và tọa độ khởi đầu của các v_i trong thuật toán FCM. Cách thực hiện như sau :

Gọi d_{mi} là độ khác biệt trung bình so với tâm lớp thứ i của các mẩu dữ liệu $xy_j \in D$.

$$d_{mi} = \frac{\sum_{j=1}^{ntd} u_{ij} \times d(xy_j, v_i)}{\sum_{j=1}^{ntd} u_{ij}} \quad (8)$$

Gọi $T_{mi}(xy_j)$ là tỉ lệ khác biệt giữa mẩu xy_j và phân lớp thứ i: $T_{mi}(xy_j) = \frac{d(xy_j, v_i)}{d_{mi}}$

Đặt $T_m(xy_j, V)$ là tỉ lệ khác biệt giữa xy_j và V : $T_m(xy_j, V) = \min_{v_i \in V}\{T_{mi}(xy_j)\}$ (9)

Với hằng số giới hạn tỉ lệ khác biệt τ; $\tau \# 0$ cho trước, ứng với mỗi mẩu dữ liệu xy_j, ta nói:

$xy_j \in V \Leftrightarrow T_m(xy_j, V) \le \tau$

$xy_j \notin V \Leftrightarrow T_m(xy_j, V) > \tau$

Phân lớp v_k chứa mẩu $xy_j \Leftrightarrow T_{mk}(xy_j) = T_m(xy_j, V)$ (10)

Dữ liệu học được tiếp nhận và xử lý như sau:

Nếu $xy_j \in V$ do v_k:

Cập nhật xy_j vào phân lớp v_k. Điều chỉnh v_k.

Tính lại các chỉ số thuộc lớp của các mẩu dữ liệu đã xét thuộc về v_k.

Nếu $xy_j \notin V$:

Bổ sung $v_{c+1} = xy_j$ vào V. Tính các chỉ số thuộc lớp cho vector mới:

$$u_{(c+1)t} = \begin{cases} 0 & , t < j \\ 1 & , t = j \end{cases} \qquad (11)$$

Thuật toán tiền xác định c và V

b1. $V = \{xy_1\}$

b2. $c = 1$

b3. for j = 2 to ntd do

tm = $d_m(xy_j, V)$

if tm > τ then

$c = c + 1$;

$V = V \cup \{xy_j\}$

Cập nhật u_{ct} theo (11), $t = \overline{1, j}$

else

Xác định v_k: $T_m(xy_j, v_k) = T_m(xy_j, V)$

Cập nhật u_{kt} theo (6), $t = \overline{1, j}$

Tính lại V theo công thức (5)

end if

end for

b4. Dừng thuật toán

Phối hợp hai thuật toán trên, ta có thuật toán FCM cải tiến (FCMm).

Thuật toán FCMm

b1. Thực hiện *Thuật toán tiền xác định c và V*

b2. Thực hiện *Thuật toán FCM*

b3. Kết thúc

2.2. *Thuật toán vector lượng tử thích nghi AVQ (Adaptive Vector Quantization)* [1]

Thuật toán AVQ sử dụng hai giá trị rõ 0,1 cho chỉ số thuộc lớp. Với mỗi mẩu dữ liệu học xy_j:

$u_{ij} = 1$, xy_i thuộc phân lớp vector v_i

$u_{ij} = 0$, xy_i không thuộc phân lớp vector v_i

Quá trình phân lớp được tiến hành bằng cách sử dụng tập vector $\{v_i\}_c$ dò trong không gian dữ liệu. Mỗi mẩu dữ liệu xy_j sẽ bị hút bởi duy nhất một vector v_k gần nhất theo qui luật cạnh tranh.

$$d(xy_j, v_k) = \min_{i=1,c}\{d(xy_j, v_i)\} \qquad (12)$$

Trọng tâm v_k của "Đám mây" các mẩu dữ liệu thuộc phân lớp v_k sẽ dao động theo hướng dịch về phía vector xy_j. Mô hình cập nhật vector v_k tại thời điểm t theo qui tắc cạnh tranh như sau :

$$v_k(t+1) = v_k(t) - \mu_t.[v_k(t) - xy_j] \qquad (13)$$

Mô hình học (13) đảm bảo các v_i hội tụ về tâm của phân lớp dữ liệu mà nó biểu diễn.

Sau đây là phần chứng minh

Mô hình học tổng quát của (13) như sau:

$$v_{kj}(t+1) = v_{kj}(t) - \mu_t.\Delta v_{kj}(t)$$

$$\text{với : } \Delta v_{kj}(t) = u_{ij}(t).[v_i(t) - xy_j] \qquad (14)$$

Tại thời điểm hội tụ t_0 của thuật toán, V đạt trạng thái cân bằng.

$\Delta v_{kj} = 0$, $\forall i = \overline{1,c}$, $\forall j = \overline{1,ntd}$, $\forall t \geq t_0$.

Tính giá trị kỳ vọng hai vế của đẳng thức (14), $\forall i = \overline{1,c}$, $\forall t \geq t_0$, ta có:

$$E[u_{ij}.[v_i - xy_j]] = 0 \ , \ \forall j = \overline{1,ntd} \qquad (15)$$

Gọi $p(xy_j)$ là hàm mật độ xác suất của dữ liệu trên các phân lớp.

$$(15) \Leftrightarrow \int_D u_{ij}[v_i - xy_j].p(xy_j)dxy_j = 0$$

$$\Leftrightarrow \int_{D_i} [v_i - xy_j].p(xy_j)dxy_j = 0$$

Suy ra : (16)

$$\int_{D_i} v_i.p(xy_j)dxy_j = \int_{D_i} xy_i.p(xy_j)dxy_j$$

Gọi v_i^0 là điểm cân bằng của v_i, v_i^0 là nghiệm của (16). Ta được:

$$v_i^0 = \frac{\int_{D_i} xy_j.p(xy_j)dxy_j}{\int_{D_i} p(xy_j)dxy_j} \qquad (17)$$

Phương trình (17) chứng tỏ v_i^0 là trọng tâm của phân lớp thứ i (D_i) của dữ liệu. (ĐPCM)

Thuật toán AVQ đồng thời xác định ma trận hiệp phương sai I theo mô hình cập nhật sau :

$$I_k(t+1)=I_i(t) + \nu_t[(xy_j-v_i(t))(xy_j-v_i(t))^T - I_k(t)] \qquad (18)$$

Ma trận nghịch đảo của I cung cấp các thông số cho việc kiến tạo khối mờ hình ellipse trong không gian dữ liệu được Bark Kosko [1] sử dụng trong hệ mờ với khối mờ hình ellipse. Phương pháp tiếp cận trong bài báo này không sử dụng ma trận I.

Thuật toán AVQ

b1. Khởi động các v_i tùy ý , $i = \overline{1,c}$. Đặt $K_i = 0$, $i = \overline{1,c}$

b2. for j = 1 to *ntd* do

Xác định v_k theo (12)

với $\mu_j = 1/j$

Tính v_k theo (13)

$\nu_t = 0.2[1 - 1/(1.2 \times ntd)]$

Tính I_k theo (18)

end for

b3. Dừng thuật toán.

Cải tiến thuật toán

Thuật toán AVQ vừa trình bày ở trên có hai vấn đề trở ngại:

- Số phân lớp c phải xác định trước
- Khởi tạo ngẫu nhiên V có thể phát sinh hiện tượng thắng cuộc liên tục của một vector v_k cố định. Điều đó buộc thuật toán mất rất nhiều thời gian lặp lại quá trình phân lớp trên dữ liệu để đảm bảo chất lượng của quá trình phân lớp.

Ứng dụng các ý tưởng cải tiến đã thực hiện với thuật toán FCM cho thuật toán AVQ. Nội dung thực hiện của thuật toán có thể tóm tắt như sau:

- Ấn định trước khoảng cách tối thiểu ε giữa hai phân lớp.
- Ứng với mỗi mẫu dữ liệu xy_j, xác định vector thắng cuộc v_k theo (12).

- Nếu $d(xy_j, v_k) \le \varepsilon$, cập nhật v_k theo công thức (13), cập nhật I_k theo công thức (18).

  Ngược lại, bổ sung $v_{c+1} = xy_j$ vào V, đặt $I_{c+1} = 0$.

Thuật toán AVQ cải tiến (AVQm)

b1. $V = \{xy_1\}$, $K_1 = 0$, $c = 1$.

b2. for j = 1 to *ntd* do

Xác định v_k theo (12)

If ($\|xy_j - v_k\| \le \varepsilon$) then

Đặt $\mu_j = 1/j$

Tính v_k theo (13)

Đặt $v_t = 0.2[1 - 1/(1.2 \times ntd)]$

Tính I_k theo (18)

Else

$c = c + 1$

$V = V \cup \{xy_j\}$

$I_c = 0$

End if

end for

b3. Tính giá trị hàm mục tiêu J(d, V)

b4. Nếu J(d, V) chưa đạt thì quay lại b2.

b5. Dừng thuật toán.

3. Ứng dụng thuật toán phân lớp dữ liệu cho học cấu trúc của hệ luật mờ SAM

3.1. *Hệ mờ SAM*

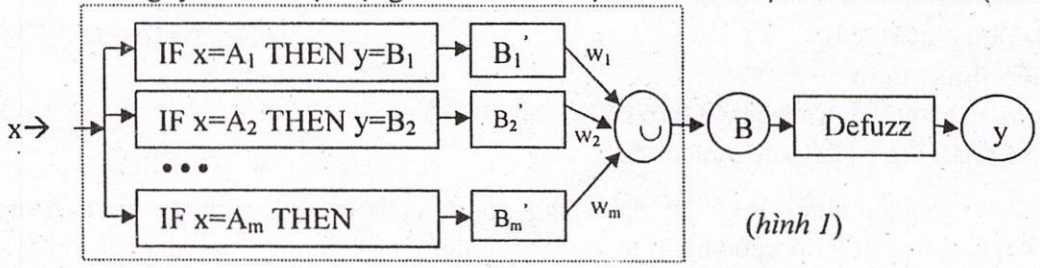
Hệ mờ SAM [1] là hệ thống m luật mờ hoạt động song song và kết hợp theo mô hình cộng-tích. Gọi R_j là luật mờ thứ j của hệ SAM. Ta có:

R_j: IF $X = A_j$ THEN $Y = B_j$, $j = \overline{1,m}$

A_j : Tập mờ đầu vào. A_j có thể là tập hợp n tập mờ thành phần A_j^i, $i = \overline{1,n}$, tương ứng với n chiều của không gian vào của luật.

B_j : Tập mờ đầu ra. B_j có thể có nhiều tập mờ thành phần. Để đơn giản cho việc tính toán và cài đặt, ta giả sử B_j có một thành phần. Như vậy số chiều của không gian ra của luật là 1.

Kiến trúc và nguyên tắc hoạt động của mô hình hệ mờ SAM được biểu diễn như (*hình 1*)



(*hình 1*)

x : Giá trị vào, $x \in R^n$.

y_0 : Giá trị ra của hệ thống, $y_0 \in R$.

A : Giá trị mờ hóa của x.

B_j' : Tập mờ kết quả cho bởi luật R_j.

w_j : Trọng số luật R_j trong hệ mờ SAM.

$\cup$: Mô hình kết hợp các tập mờ kết quả B_j' của các luật mờ trong hệ SAM. $\cup$ của SAM là *cộng* (sum). Ở một số mô hình khác, $\cup$ có thể là *lớn nhất* (max).

B : Tập mờ kết quả của hệ thống.

Với mỗi giá trị vào x, x được mờ hóa và kích hoạt tất cả các luật. Gọi $a_j(x)$ là mức kích hoạt luật R_j:

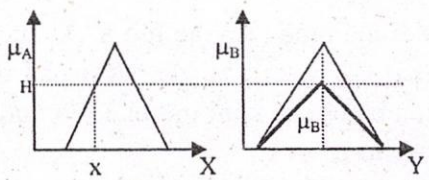
$$a_j(x) = \prod_{i=1}^{n} a_j^i(x_i) \tag{19}$$

$a_j^i(x_i)$ là giá trị hàm liên thuộc mờ của tập mờ thành phần A_j^i trong A_j.

Giá trị tập mờ đầu ra của R_j là :

$$B_j' = a_j(x).B_j \tag{20}$$

Ví dụ: *xét luật R_j. (Hình 2) sử dụng đường đơn biểu diễn tập mờ vế trái A_j và vế phải B_j. Giá trị x kích hoạt vế trái luật $H = \mu_A(x)$. Tập mờ kết quả B' có μ_B. được xác định bằng đường nét đậm.*



(hình 2)

Tập mờ đầu ra B của hệ thống:

$$B = \sum_{j=1}^{m} w_j.B_j' = \sum_{j=1}^{m} w_j.a_j(x).B_j \tag{21}$$

Đặt :

b : Hàm liên thuộc mờ của tập mờ B.

b_j : Hàm liên thuộc mờ của tập mờ B_j, $j = \overline{1,m}$.

b_j' : Hàm liên thuộc mờ của tập mờ B_j', $j = \overline{1,m}$.

V_j : Kích thước khối mờ B_j.

c_j : Trọng tâm khối mờ B_j.

Ta có:

$$V_j = \int_R b_j(y)dy \tag{22}$$

$$c_j = \frac{\int_R y.b_j(y)dy}{\int_R b_j(y)dy} \tag{23}$$

$$b(y) = \sum_{j=1}^{m} w_j.b_j'(y) = \sum_{j=1}^{m} w_j.a_j(x).b_j(y) \tag{24}$$

Sử dụng phương pháp khử mờ trọng tâm [3] đối với tập mờ B để xác định giá trị ra y_0 :

$$y_0 = \text{Centroid}(B) = \frac{\int_R y.b(y)dy}{\int_R b(y)dy} = \frac{\int_R y.\sum_{j=1}^{m} w_j.b_j'(y)dy}{\int_R \sum_{i=1}^{m} w_i.b_i'(y)dy} = \frac{\int_R y.\sum_{j=1}^{m} w_j.a_j(x).b_j(y)dy}{\int_R \sum_{i=1}^{m} w_i.a_i(x).b_i(y)dy}$$

$$= \frac{\sum_{j=1}^{m} w_j.a_j(x).\int_R y.b_j(y)dy}{\sum_{i=1}^{m} w_i.a_i(x).\int_R b_i(y)dy} = \frac{\sum_{j=1}^{m} w_j.a_j(x).V_j \frac{\int_R y.b_j(y)dy}{\int_R b_j(y)dy}}{\sum_{i=1}^{m} w_i.a_i(x).V_i} \quad (25)$$

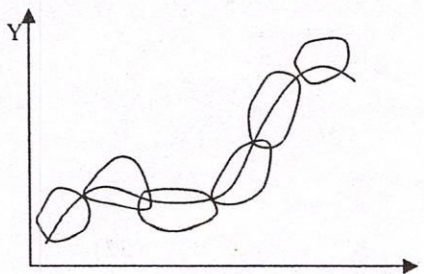
Mô hình hệ mờ SAM: $y = F(x) : R^n \rightarrow R$

$$F(x) = \frac{\sum_{j=1}^{m} w_j.a_j(x).V_j.c_j}{\sum_{i=1}^{m} w_i.a_i(x).V_i} \quad (26)$$

Với mô hình (26), hệ mờ SAM cho phép xấp xỉ mô hình tương quan giữa 2 đại lượng X,Y. Chất lượng xấp xỉ của SAM phụ thuộc vào số luật có trong SAM và được thể hiện trực quan bằng các khối mờ phủ lên đường biểu diễn của quan hệ xấp xỉ trong không gian XY (*hình 3a & 3b*) .

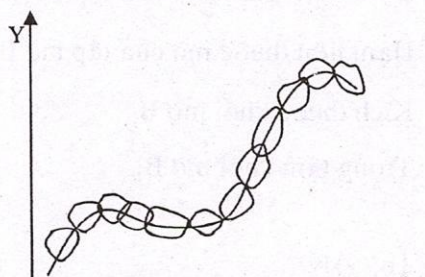
Ngoài số lượng luật, cách phân bố của các khối mờ hình thành từ các luật cũng là yếu tố quyết định đến chất lượng xấp xỉ của SAM.

Vì vậy, với bộ dữ liệu thống kê phản ánh quan hệ giữa hai đại lượng X và Y bất kỳ, để xây dựng mô hình SAM xấp xỉ tương quan giữa chúng với chất lượng cao thì phân lớp dữ liệu làviệc giải quyết trước tiên. Các thuật toán phân lớp cải tiến đã trình bày trong phần 2 giúp thực hiện công việc này. Sau khi các khối mờ trong không gian XY được xác định, ta thực



(hình 3a)

SAM có số lượng khối mờ ít hơn



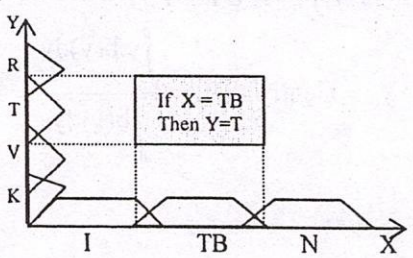
(hình 3b)

SAM có số lượng khối mờ nhiều

hiện tạo các luật mờ cho hệ SAM. Ứng với mỗi khối mờ là một luật mờ được tạo thành (*hình 4*).TB - tập mờ đầu vào, T - Tập mờ đầu ra. Ta có luật: IF X = TB THEN Y = T

3.2. *Dạng hàm liên thuộc mờ*

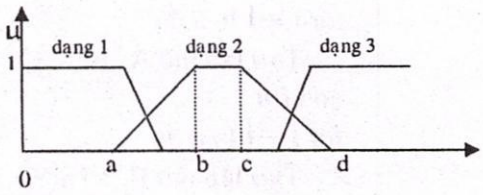
Mỗi khối mờ trong không gian XY được chuyển thành luật mờ với các tập mờ vế trái là hình chiếu của khối mờ lên trục X (vào) và tập mờ vế phải là hình chiếu của khối mờ lên trục Y (ra) . Đường biểu diễn hàm liên thuộc mờ của các tập mờ này có tâm tại điểm chiếu tâm khối mờ, và có nhiều dạng biến thiên khác nhau. Dạng biến thiên của các hàm liên thuộc mờ được bài báo sử dụng là dạng hình thang.



Với dạng hình thang, dáng điệu của hàm liên thuộc mờ chỉ là một trong các dạng như (*hình 5*).

Khi đó, hàm liên thuộc mờ có thể được biểu diễn thông qua 4 chỉ số : a, b, c, d xác định hoành độ 4 đỉnh của hình thang biểu diễn hàm.

(*hình 4*)

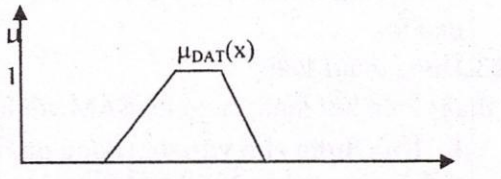


(*hình 5*)

Ví dụ: Cho cơ sở rõ P = [0, ∞) các chỉ số giá của một mặt hàng, và khái niệm ngôn ngữ "ĐẮT" xác định trên cơ sở này. Mức độ liên thuộc được cho rằng hàng hóa chỉ hoàn toàn đắt (100%) với mức giá là 30000-40000, giảm đều về hai phía và bằng 0 khi giá hàng dưới 15000 hoặc trên 45000. Hàm μ_{DAT} phản ánh mức độ liên thuộc mờ ĐẮTđược biểu diễn như sau:

$$\mu_{DAT} : [0, \infty) \rightarrow [0,1] \quad \forall p \in [0, \infty) :$$

$$\mu_{DAT}(p) = \begin{cases} 1 & 30000 \le p \le 40000 \\ \dfrac{p - 15000}{15000} & -15000 < p < 30000 \\ \dfrac{45000 - p}{5000} & -40000 < p < 45000 \\ 0 & p \le 15000, p \ge 45000 \end{cases}$$



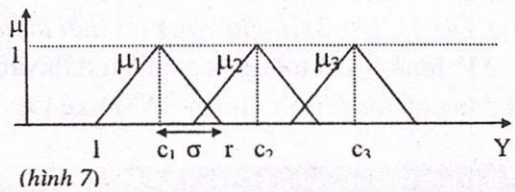
(*hình 6*)

Đồ thị hàm liên thuộc mờ μ_{DAT} (*hình 6*) Ta viết : μ_{DAT}(15,30, 40, 45)

3.3. *Xây dựng kiến trúc hệ SAM từ kết quả của quá trình phân lớp dữ liệu*

Các thuật toán phân lớp dữ liệu giúp xác định số phân lớp và trọng tâm v_i của chúng. Từ các phân lớp này ta có thể kiến tạo các luật mờ cho hệ mờ SAM.

Mỗi v_i xác định một khối mờ trong không gian XY của mô hình cần xấp xỉ Y = f(X). Chiếu v_i lên các trục của không gian XY, ta được trong tâm của các tập mờ. Trên mỗi trục, dựa vào các trọng tâm tập mờ, ta xây dựng các tập mờ. Trọng tâm tập mờ được chọn làm hoành độ đỉnh tập mờ; chân tập mờ được xác định tại điểm giao giữa tập mờ ấy với tập mờ lân cận thông qua hệ số chồng lấn cho trước θ. Gọi σ là độ rộng chân tập mờ tính từ tâm đỉnh [3]:



(*hình 7*)

$$\sigma = \frac{|m - m_{closest}|}{\theta}$$

$m_{closest}$: Tâm tập mờ lân cận gần nhất.

Giả sử Y trong không gian XY có 3 trọng tâm c_1, c_2, c_3. Ba tập mờ tương ứng có đường biểu diễn như (*hình 7*).

Trong đó, μ_1 được biểu diễn: μ_1(l, c_1, c_1, r)

Sau khi hoàn tất việc tạo tập mờ trên các trục, dựa vào các khối mờ tương ứng với các phân lớp, ta tạo ra luật mờ cho hệ SAM (hình 4).

Trọng số luật : Các luật mờ đưa vào hệ SAM có tầm quan trọng khác nhau đối với vai trò xấp xỉ của hệ. Tầm quan trọng đó được thể hiện bằng trọng số của luật. Trọng số luật R_j hình thành trong hệ mờ từ phân lớp được xác định như sau:

$$w_j = \frac{|D_j|}{ntd}$$, D_j là phân lớp ứng với v_j. (27)

Thuật toán tạo cấu hệ SAM từ V

b1. Khởi tạo hệ SAM rỗng

b2. for j = 1 to c do
for i = 1 to n do
Tạo tập mờ A_j^i trên X^i.
end for
for i = 1 to p do
Tạo tập mờ B_j^i trên Y^i.
end for
Tính w_j theo (27)
Đặt R_j : $\{A_j^i\} \rightarrow B_j$.
Bổ sung $<w_j, R_j>$ vào SAM
end for
b3. Dừng thuật toán

Khi thuật toán kết thúc, ta có hệ SAM với hệ thống c luật mờ.

4. Ứng dụng cho vấn đề tương quan và dự báo

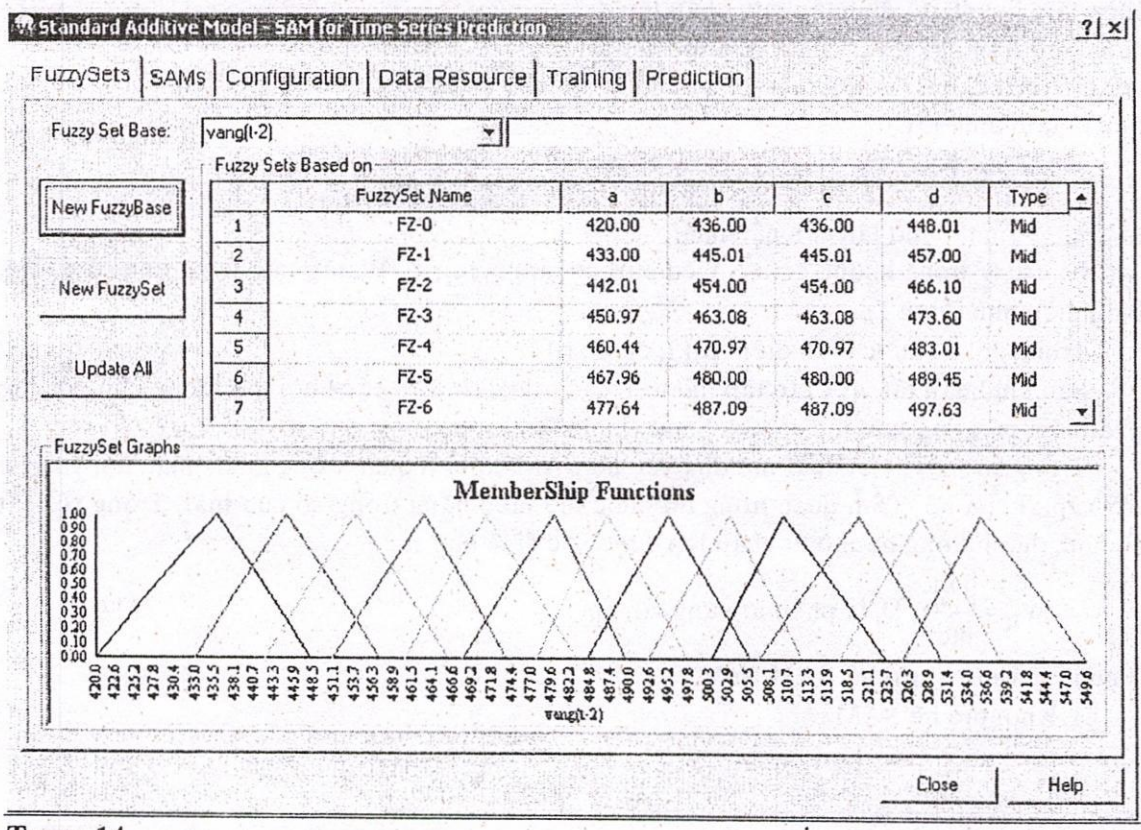
Phần mềm "Ứng dụng SAM cho dự báo chuỗi thời gian" [9] là sản phẩm thực hiện dựa trên các nội dung trình bày trong bài báo này. Với bộ dữ liệu có 2327 mẩu về giá vàng/ngày trong các năm 93-97 và 99-2000 tại tp.HCM, phần mềm cho phép tạo hệ SAM xấp xỉ tương quan giữa các đại lượng trong mô hình dự báo giá vàng mà các chuyên gia Viện Kinh Tế tp.HCM đã xây dựng bằng phương pháp Arima (kinh tế lượng), từ đó có thể thay thế mô hình kinh tế lượng bằng mô hình SAM.

- Mô hình kinh tế lượng:

$D(t) = 0.139 + 0.056[D(t-1) - D(t-2)] \pm 0.262 + D(t-1)$

D(t), D(t-1), D(t-2) là giá vàng tại thời điểm khảo sát, trước thời điểm khảo sát 1và 2 ngày.

- Mô hình SAM tương ứng có hai đầu vào cho D(t-2) và D(t-1) và một đầu ra cho D(t). Sử dụng chương trình Times-SAM.exe [9].



❖*Khai báo kiến trúc hệ SAM* : Chọn ngăn ***SAMs***

Chọn đầu ra của hệ SAM (Output Base) là vang(t).

Chọn các đầu vào (Input Bases) là vang(t-2) và vang(t-1).

❖*Huấn luyện SAM bằng thuật toán học cấu trúc*

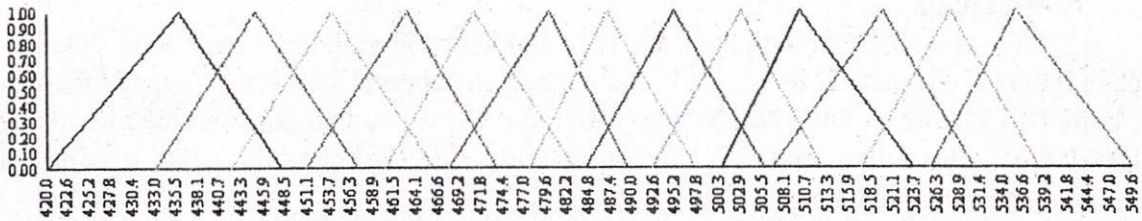
Sau khi dùng ngăn ***Data Resource*** để đọc dữ liệu học và dữ liệu thử nghiệm [9], trong ngăn ***Configuration***, Chọn hệ SAM cần huấn luyện, sau đó ấn định các thông số học cấu trúc.

- Chọn mục học cấu trúc (Structure Learning).
- Chọn phương pháp phân lớp dữ liệu - giả sử ta sử dụng phương pháp AVQ
- Khai báo số phân lớp ban đầu c. Đặt c = 0 để áp dụng thuật toán AVQ cải tiến.

Ấn định khoảng cách d giữa các phân lớp (chỉ dùng với thuật toán cải tiến).

Để kiểm chứng, chúng tôi tạo ngẫu nhiên bộ dữ liệu test từ dữ liệu thống kê nói trên theo mô hình 2 đầu vào-1 đầu ra (tập tin vangtrain.AID [9]). Với chiều dài dự báo là 30 ngày, kết quả như sau:

❖*Với d = 7* : Hệ SAM có 51 luật mờ, Các tập mờ hình thành trên cơ sở đầu ra là *vang(t)* như sau.



Áp dụng hệ SAM cho dự báo cùng với mô hình kinh tế lượng, ta có chỉ số so sánh như (*bảng 1*).

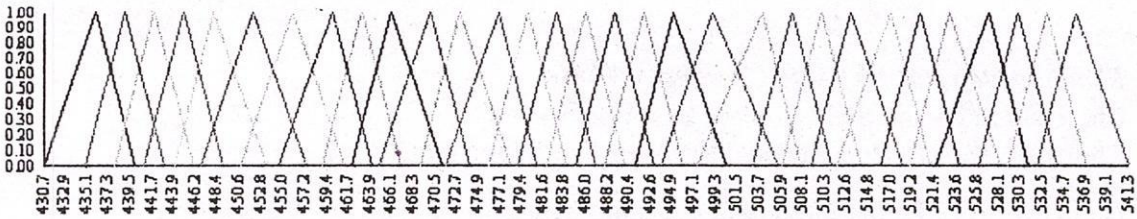
Bảng 1

Các chỉ số thống kê	SAM	Kinh tế lượng	
		+0.262	-0.262
Giá trị lớn nhất	494.19	494.32	494.85
Giá trị nhỏ nhất	480.10	479.82	480.35
Giá trị trung bình	484.30	484.80	485.33
Sai số trung bình	2.10	1.96	1.91
Sai số phần trăm trung bình	0.43	0.40	0.39
Sai số Tbình bình phương	7.77	6.81	6.47
Giá trị lớn nhất	494.19	+0.262	-0.262

Bảng 2

Các chỉ số thống kê	SAM	Kinh tế lượng	
		+0.262	-0.262
Giá trị lớn nhất	492.99	494.32	494.85
Giá trị nhỏ nhất	481.04	479.82	480.35
Giá trị trung bình	484.61	484.80	485.33
Sai số trung bình	1.91	1.96	1.91
Sai số phần trăm trung bình	0.39	0.40	0.39
Sai số Tbình bình phương	6.32	6.81	6.47
Giá trị lớn nhất	492.99	+0.262	-0.262

❖*Với d = 3* : Hệ SAM có 190 luật mờ, Các tập mờ hình thành trên cơ sở đầu ra là *vang(t)* như sau.



Chỉ số so sánh của hệ với các mô hình kinh tế lượng được ghi trong (*bảng 2*).

Nhận xét: Trong phần minh họa nói trên, chúng tôi sử dụng 2 giá trị khác nhau của d (d chính là τ trong FCMm hay ε của AVQm). Với d lớn, tức là nới lỏng ràng buộc về tính đồng nhất của các mẩu trong cùng phân lớp, ta nhận được số phân lớp ít, dẫn đến số luật sinh ra ít. Khi d nhỏ, chất lượng đồng nhất của các mẩu trong cùng phân lớp sẽ tăng lên, và vì vậy số phân lớp cũng tăng theo, dẫn đến số luật sinh ra nhiều hơn (khi d giảm từ 7 xuống 3, số luật tăng lên 4 lần). Tuy vậy, d nhỏ sẽ giúp cho các khối mờ tập trung về tâm của đám mây dữ liệu, nhờ đó mà chất lượng xấp xỉ của hệ mờ được nâng cao. Với d chọn bằng 7, sai số trung bình của hệ mờ là 2.10 (xem *bảng 1*) kém hơn các mô hình toán. Nhưng khi điều chỉnh với d bằng 3, sai số chỉ còn là 1.9 (xem *bảng 2*), tốt hơn các mô hình toán. Điều này cũng giúp khẳng định quan điểm đã trình bày trong mục 3.1 của bài báo.

5. Kết luận.

Việc sử dụng thuật toán phân lớp cải tiến không chỉ giúp quá trình học tri thức của phần mềm [9] đạt mức độ tự động hóa cao mà còn cho phép phần mềm đáp ứng nhiều nhu cầu dự báo. Chúng tôi cũng đã áp dụng phần mềm này để dự báo giá của nhiều loại hàng hóa đặc biệt như dollar, thép, gạo, ciment và đã đạt được kết quả tốt hơn khi so sánh với các mô hình toán như đề cập ở trên.

Vấn đề mà chúng tôi sẽ tiếp tục nghiên cứu là xây dựng thuật toán xác định hệ số d cho bài toán phân lớp dựa trên giá trị sai số cho phép của mô hình dự báo. Đây là vấn đề khá mới mẻ và chưa có công trình nghiên cứu nào về vấn đề này được công bố. Rút kinh nghiệm từ việc phát triển ứng dụng [9], chúng tôi sẽ hoàn chỉnh thêm giải pháp cho bài toán đã được trình bày trong mục 2 của bài báo này theo hướng phục vụ cho các mô hình máy học làm chức năng xấp xỉ hàm phi tuyến, đồng thời áp dụng những kết quả đạt được vào phần mềm trò chơi kinh doanh mà hiện nay chúng tôi đang thực hiện.

IMPROVEMENT OF THE TWO DATA CLUSTERING ALGORITHMS: FCM & AVQ AND APPLICATION TO APPROXIMATE FUZZY SYSTEM

Le Ngoc Thanh

ABSTRACT: *Knowledge discovery is one of the most important problems in the knowledge engeneering domain. Where, data clustering algorithms take the prerequisite role to the quality of received knowledge. This paper presents some improments in the two data clustering algorithms FCM and AVQ. Then there is, in the same context, also an applying of the research result to the unsupervisor learning phase of the SAM system of correlational approximation.*

Keywords: *Data Clustering, Fuzzy C-Means Algorithm, Adaptive Vector Quantization Algorithm, Fuzzy System, Standard Additive Model.*

TÀI LIỆU THAM KHẢO

[1] Bart Kosko, *Fuzzy Engineering*, Prentice Hall International Inc. 1997.

[2] GSTS.Hoàng Kiếm, TS.Vũ Thanh Nguyên, Nhóm nghiên cứu, *Phân Tích Dự Báo Kinh Tế Ứng Dụng Trong Ngành Công Nghiệp Tại Tp Hcm*, Sở KHCN Tp.HCM 2003.

[3] Chin-Teng Lin and C.S. George Lee, *Neural Fuzzy System*, Prentice Hall 1996.

[4] Parag M. Kanade and Lawrence O. Hall, *Fuzzy Ant Clustering by Centroid Positioning*, (2004). www.csee.usf.edu/~hall/papers/antcluscent.pdf.

[5] Carla S. Moller-Levet, Kwang-Hyun Cho, *Data Clustering Based on Temporal Variation: FCV with TSD Preclustering* (2003). http://images.ee.umist.ac.uk/carla/FCV_TSD_in_AB.pdf.

[6] A. M. Bagirov, A. M. Rubinov, *Unsupervised and supervised data classification via nonsmooth and global optimization* (2003).

www.mat.univie.ac.at/~neum/glopt/mss/BagRS03.pdf.

[7] N. Hussein, *A Fast Greedy K-Means Algorithm* (2002),

http://www.science.uva.nl/research/ias/alumni/m.sc.theses/theses/NoahLaith.doc

[8] Lê Ngọc Thạnh, *Ứng dụng logic mờ cho Mô hình thực tế ảo Thị trường các Doanh nghiệp cạnh tranh và Phần mềm Trò Chơi Kinh Doanh*, Luận văn Thạc sĩ Công Nghệ Thông Tin - Trường Đại Học Khoa Học Tự Nhiên, Đại Học Quốc Gia Thành phố Hồ Chí Minh (2000).

[9] Lê Ngọc Thạnh, *Phần mềm Ứng dụng mô hình SAM cho dự báo chuỗi thời gian, Đề Tài Phân Tích Dự Báo Kinh Tế Ứng Dụng Trong Ngành Công Nghiệp Tại Tp Hcm*, Sở KHCN Tp.HCM (2003). http://thanh.andisw.com/MachineLearning/Times-SAM.exe.