

ỨNG DỤNG DỰ BÁO BIỂU ĐIỂM CHO ĐỀ THI TRẮC NGHIỆM KHÁCH QUAN BẰNG HỆ LUẬT MỜ

• ThS. Lê Duy Đồng^(*); PGS, TS. Vũ Thanh Nguyên^(**)
KS. Lê Kim Nga^(***)

Tóm tắt

Bài báo này đề xuất một phương pháp dự báo biểu điểm cho đề thi trắc nghiệm khách quan bằng hệ luật mờ Standard Additive Model (SAM). Ngày nay, việc ứng dụng thành tựu khoa học máy tính vào ngành Khảo thí và Kiểm định chất lượng giáo dục vẫn còn nhiều hạn chế. Chúng tôi đã nghiên cứu và ứng dụng thành công SAM vào việc dự báo biểu điểm cho đề thi trắc nghiệm khách quan. Qua thực nghiệm, ứng dụng cho kết quả khả quan.

Từ khóa: Hệ luật mờ, dự báo dữ liệu, trắc nghiệm.

1. Đặt vấn đề

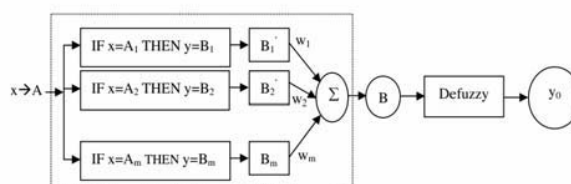
Từ khi con người phát minh ra máy tính, người ta luôn tìm cách để nó không những thu nhận và xử lý dữ liệu mà còn đưa ra những dự báo dựa trên những dữ liệu đó. Ngày nay, khoa học máy tính đã phát triển vượt bậc. Những giải thuật tiên tiến đã giúp máy tính có khả năng dự báo dữ liệu trong nhiều ngành như: kinh tế [4][8], khí tượng thủy văn [9]... Vì vậy, máy tính có thể giúp con người đưa ra những dự báo cho tương lai.

Ở kỳ thi tuyển sinh Cao đẳng, Đại học năm 2011, môn Sử có rất nhiều bài thi bị điểm 0 và trở thành đề tài nóng bỏng của dư luận xã hội [7]. Theo phỏng vấn của Dân Trí với GS, TS Đỗ Thanh Bình, Chủ nhiệm khoa Lịch sử, Trường Đại học Sư phạm Hà Nội thì việc này có nhiều nguyên nhân nhưng chủ yếu là do đề thi và đáp án có vấn đề. Như vậy, một đề thi không phải lúc nào cũng sát với học lực của người học.

Từ những thực tiễn trên, là những giảng viên, chuyên viên, giảng dạy và tham gia công tác Khảo thí và Kiểm định chất lượng giáo dục (KT&KĐCLGD). Chúng tôi luôn trăn trở tìm kiếm phương thức giúp giảng viên chọn đề thi phù hợp với học lực người học. Qua quá trình

nghiên cứu, chúng tôi đã ứng dụng thành công SAM vào dự báo biểu điểm dựa trên cấu trúc độ khó của đề thi và học lực của người học. Từ đó góp phần cho việc đánh giá kết quả học tập của người học đạt hiệu quả và khoa học hơn.

2. Mô tả SAM



Hình 1. Mô tả các thành phần của mô hình SAM [4], [6], [8]

Trong đó:

x : Giá trị vào, $x \in R_n$.

y_0 : Giá trị ra của hệ thống $= F(x_0)$

A_j : Giá trị mờ hóa của x thứ j .

B_j : Tập mờ kết quả cho bởi luật R_j .

w_j : Trọng số của luật R_j trong hệ luật.

#: Qui tắc cộng trong mô hình luật kết hợp cộng tích (SUM - PRODUCT).

B : Tập mờ kết quả của toàn hệ luật.

Hệ luật mờ là hệ thống m luật mờ dạng R_j :
IF $x = A_j$ THEN $y = B_j$ $j = \overline{1, m}$, hoạt động theo cơ chế song song. Có nhiều cách kết hợp vế trái và vế phải trong mỗi luật mờ và kết hợp kết quả của các luật mờ trong hệ luật. Tuy nhiên, vì các đặc tính thuận lợi trong tính toán (tính tích phân để xác định trọng tâm), bài báo này chỉ đề cập

^(*) Khoa Công nghệ thông tin, Trường Cao đẳng Kinh tế Tài chính Vĩnh Long.

^(**) Khoa Công nghệ phần mềm, Trường Đại học Công nghệ Thông tin.

^(***) Bộ môn Tin học, Trường Trung học phổ thông Vĩnh Long.

đến hệ luật mờ hoạt động theo qui tắc kết hợp SUM-PRODUCT. Do tính kết hợp SUM các luật mờ mà hệ luật mờ này có tên gọi là SAM (Standard Additive Model). Trong hệ mờ SAM, ứng với mỗi giá trị vào $x=x_0$, luật thứ j : R_j được kích hoạt và cho kết quả là tập mờ B_j' xác định theo B_j và mức độ thỏa mãn về trái $a_j(x_0)$ dựa trên qui tắc PRODUCT.

$$B_j' = a_j(x_0) \cdot B_j \quad [4] \quad (1)$$

Với $a_j(x_0)$ là mức kích hoạt luật R_j . Và $a_j(x)$ được cho bởi công thức

$$a_j(x) = \prod_{i=1}^n a_j^i(x_i) \quad [4]$$

m kết quả ra B_j' của các luật trong hệ luật được SAM kết hợp theo qui tắc SUM để cho kết quả chung của toàn hệ thống là tập mờ B.

$$B = \sum_{j=1}^m w_j \cdot B_j' = \sum_{j=1}^m w_j \cdot a_j(x_0) \cdot B_j \quad [4] \quad (2)$$

Giá trị B sẽ được khử mờ để nhận được một giá trị rõ duy nhất. Gọi y là kết quả sau khi khử mờ tập mờ kết quả B của hệ SAM, ta có:

$$y = F(x_0) = \text{Centroid}(B(x_0)) = \frac{\sum_{j=1}^m w_j \cdot a_j(x_0) \cdot V_j \cdot c_j}{\sum_{j=1}^m w_j \cdot a_j(x_0) \cdot V_j} \quad [4] \quad (3)$$

Với Centroid là phương pháp khử mờ trọng tâm.

3. Cơ chế học trong SAM

Học là hoạt động quan trọng của SAM nhằm xây dựng cơ sở tri thức cần thiết dưới dạng các luật mờ phục vụ cho việc mô phỏng hàm phi tuyến mà hệ đảm nhận. Nhờ cơ chế học mà SAM có thể khắc phục hạn chế là phải chờ đợi tri thức chuyên gia. Ngoài ra, việc học còn giúp cho SAM có khả năng tự điều chỉnh các thông số cấu trúc cũng như kiểm tra tính tối ưu trong cấu trúc luật để từ đó có thể nâng cao độ chính xác trong hoạt động xấp xỉ của nó. Thông qua việc hiệu chỉnh các thông số: kích thước, trọng tâm các tập mờ, SAM tự động định vị lại vị trí và kích thước của các khối mờ, nhờ đó quá

trình xấp xỉ của SAM được chính xác hơn. Đối với SAM, một phương pháp học được đánh giá là tốt hay không ở chỗ nó có điều chỉnh nhanh chóng hay không kích thước và vị trí của các khối mờ cũng như đảm bảo duy trì các khối mờ tại các điểm uốn của hàm f.

Quá trình học của SAM (hay của hệ mờ nói chung) thông thường bao gồm hai bước chính là học cấu trúc và học tham số. Tuy nhiên, để cho hiệu quả học của hệ được tốt hơn, chúng tôi phối hợp thêm cơ chế học tối ưu hệ luật. Do đó, quá trình học của SAM ở đây bao gồm các giai đoạn sau:

- Tự phát sinh cấu trúc luật (structure learning): Đây là bước khởi đầu của quá trình học. Bước này được thực hiện theo cơ chế tự học (unsupervised learning). Bằng cách thực hiện việc phân cụm mờ trên bộ dữ liệu học, hệ SAM sẽ tự phát hiện ra các luật mờ cần thiết cho việc xấp xỉ hàm phi tuyến đặc trưng cho bộ số liệu học đó.

- Điều chỉnh các thông số (parameter learning): Sau khi đã xây dựng được bộ luật mờ cần thiết, công việc học tiếp theo của SAM là điều chỉnh các thông số của hệ luật như: trọng số của từng luật, trọng tâm và kích thước của các tập mờ tham gia ở về trái và về phải của các luật. Quá trình này được tiến hành theo cơ chế học có giám sát (supervised learning). Thông qua việc kiểm tra sai số giữa giá trị kết quả do hệ sinh ra từ bộ dữ liệu học với kết quả mong muốn, hệ sẽ thực hiện điều chỉnh lại các thông số cho phù hợp. Việc điều chỉnh thông số được tiến hành theo thuật toán học điều chỉnh sai số nhỏ nhất.

- Tối ưu hóa hệ luật (optimal learning): Giai đoạn học này cho phép SAM có thể phát hiện và loại bỏ các luật mờ không cần thiết cho hoạt động xấp xỉ của nó. Nhờ cơ chế này mà SAM có thể gọn nhẹ hơn nhằm tăng đáng kể tốc độ xử lý cũng như giảm nhiễu. Cơ chế học ở đây cũng gần giống như cơ chế học có giám sát.

3.1. Học cấu trúc

Giải thuật gom cụm mờ trung bình FCM (Fuzzy C-Means) [1][2]

Gom cụm mờ dữ liệu là bước quan trọng trong quá trình xây dựng các luật mờ của một hệ mờ tự học. Nhiệm vụ đặt ra đối với quá trình gom cụm mờ dữ liệu (hay gom cụm dữ liệu nói chung) là từ một tập hữu hạn các bộ số liệu học cho trước, tìm cách tách chúng ra thành từng nhóm với các đặc trưng riêng sao cho các bộ số liệu trong cùng một nhóm càng giống nhau và các bộ số liệu giữa các nhóm càng khác nhau càng tốt. Đối với quá trình gom cụm mờ dữ liệu mờ thì bên cạnh việc phân tách dữ liệu, phải tiến hành đồng thời việc xác định mức độ phụ thuộc của từng bộ số liệu đối với nhóm mà nó thuộc về, để từ đó có thể xác định hàm thành viên của tập mờ tương ứng với nhóm sẽ được tạo ra sau này.

Bài toán gom cụm mờ dữ liệu được phát biểu như sau:

Gọi R_n là không gian các vector có n thành phần thực.

Đặt $X = \{x_1, x_2, \dots, x_{ntd}\}$, $x_j \in R_n$, là tập hữu hạn bộ số liệu học, trong đó ntd (number of training data) là số bộ dữ liệu học (x_j là một dòng trong bảng 1).

Gọi V_{cn} là không gian vector các ma trận $c \times n$, $c \in Z^+$ cho trước, $1 < c < n$.

Xác định một gom cụm mờ trên X biểu diễn bởi một bộ vector trọng tâm:

$V = \{v_1, v_2, \dots, v_c\}$, $v_i \in R_n$, cho tương ứng với l ma trận

$U = \{u_{ij}\} \in V_{cn}$, với u_{ij} là giá trị thực trong đoạn $[0,1]$ diễn tả mức độ phụ thuộc của bộ số liệu học x_j ứng với vector trọng tâm v_i , và thỏa công thức (4.1), (4.2):

- Các công thức:

$$\forall x \in X, u_{ij} \in [0,1], \sum_{k=1}^c u_{ik} = 1 \quad (4.1)$$

$$\forall i, j = \overline{1, c}: \quad 0 < \sum_{j=1}^{ntd} u_{ij} < n \quad (4.2)$$

$$v_i = \frac{\sum_{j=1}^{ntd} u_{ij}^m \cdot x_j}{\sum_{j=1}^{ntd} u_{ij}^m}, \quad u_{ij} = \frac{\left(\frac{1}{\|x_j - v_i\|^2} \right)^{\frac{1}{1-m}}}{\sum_{k=1}^c \left(\frac{1}{\|x_j - v_k\|^2} \right)^{\frac{1}{1-m}}} \quad (5), (6)$$

$$\text{Error} = \max_{ij} \left(\|u_{ij}(t+1) - u_{ij}(t)\| \right) < \varepsilon$$

với $\varepsilon > 0$ cho trước, $t > t_0$.

- **Mô tả thuật toán:**

- Vào: Bộ số liệu học X .

- Ra: Bộ vector trọng tâm V và ma trận phụ thuộc U .

b1. $t=0$, Khởi tạo ngẫu nhiên giá trị $U(t)$ thỏa (4.1) và (4.2).

b2. Xác định $V(t)$ theo công thức (5).

b3. Tính $U(t+1)$ theo công thức (6).

Xác định error theo công thức (7).

b4. Nếu error $> \varepsilon$ thì $t=t+1$, quay lại bước b2.

b5. Kết thúc.

3.2. Học thông số

Khi các luật mờ đã được xác định, học điều chỉnh thông số giúp giảm sai số giữa kết quả của hệ và kết quả mong muốn. Bài toán được phát biểu như sau:

Cho trước bộ dữ liệu học vào $\{x_j\}$ và bộ kết quả mong muốn $\{y_j\}$, $j = \overline{1, ntd}$; với ntd : số bộ dữ liệu học (xem bảng 1).

Cho hệ luật mờ SAM với các luật mờ và trọng số.

Hãy điều chỉnh thông số của các tập mờ vế trái, vế phải và trọng số các luật mờ sao cho sai số giữa kết quả cho bởi hệ luật mờ và kết quả mong muốn là ổn định và nhỏ nhất.

Các công thức:

$$c_j(t+1) = c_j(t) + \mu_t \cdot \varepsilon(x) \cdot p_j(x) \quad (8)$$

$$V_j(t+1) = V_j(t) + \mu_t \cdot \varepsilon(x) \cdot [c_j - F(x)] \cdot \frac{p_j(x)}{V_j} \quad (9)$$

$$w_j(t+1) = w_j(t) + \mu_t \cdot \varepsilon(x) \cdot [c_j - F(x)] \cdot \frac{p_j(x)}{w_j} \quad (10)$$

$$a_{ji}(t+1) = a_{ji}(t) + \mu_t \cdot \varepsilon(x) [c_j - F(x)] \frac{p_j(x)}{a_j} \cdot \frac{\partial a_j}{\partial a_{ji}} \quad (11)$$

Trong đó:

c_j : Trọng tâm tập mờ thứ j

V_j : Độ rộng của tập mờ thứ j

w_j : Trọng số luật mờ thứ j

a_{ij} : Ngưỡng kích hoạt của luật mờ thứ j

$$p_j = \sum_j w_j \cdot a_j \cdot V_j$$

Mô tả thuật toán học thông số (HTS):

Vào:

- Bộ số liệu học.

- Hệ luật mờ SAM chưa điều chỉnh.

- Sai số cho phép ε .

Ra:

- Hệ luật mờ SAM đã được điều chỉnh

theo sai số ε .

b1. Bắt đầu.

b2. $j=0$.

b3. Xét giá trị vào kế tiếp x_j . Tính $y_j =$

$$F(x_j), \varepsilon = y_j^d - y_j$$

b4. Điều chỉnh các trọng số w_j của các luật theo công thức (10).

b5. Lặp lại các bước 3,4,5.

b6. Thực hiện các bước 2,...,5 cho các B_j theo công thức (8), (9).

b7. Thực hiện các bước 2,...,5 cho các A_{ji} theo công thức (11).

b8. Tính sai số $\text{error} = E(t) - E(t-1)$.

Trong đó $E(t)$ và $E(t-1)$ được xác định theo công thức sau:

$$E(t) = \frac{1}{2} [y(t) - y^d(t)]^2$$

b9. Nếu $\text{error} > \varepsilon$ thì quay lại b1. Ngược lại dừng thuật toán.

3.3. Học tối ưu

Về mặt lý thuyết, một hệ SAM càng có nhiều luật thì độ chính xác trong hoạt động xấp xỉ của nó càng lớn. Tuy nhiên, nếu hệ có quá nhiều luật thì thời gian cho quá trình xử lý trong hệ luật sẽ là yếu tố đáng quan tâm. Một vấn đề đặt ra là làm sao có thể giải quyết hợp lý mối quan hệ giữa kích thước (số luật) của hệ SAM và độ chính xác trong xấp xỉ của nó. Một hệ luật tối ưu sẽ chỉ giữ lại một số (hoặc tất cả) các luật ban đầu trong hệ SAM. Như vậy, trong số các luật của SAM, sẽ có một số luật bị loại và một số luật được giữ lại.

Bài toán tối ưu hệ luật được phát biểu như sau:

Cho trước bộ dữ liệu học vào $\{x_j\}$ và bộ kết quả mong muốn $\{y_j\}$, $j=1, ntd$; với ntd : số bộ dữ liệu học.

Cho hệ luật mờ SAM với các luật mờ và trọng số.

Hãy tìm tập hợp các luật ít nhất sao cho sai số giữa kết quả cho bởi hệ SAM và kết quả mong muốn là ổn định và nhỏ nhất.

Một trong số các giải pháp cho bài toán trên là phương pháp sử dụng thuật toán di truyền [5]. Phương pháp này xem mỗi bộ kết hợp các luật là một nhiễm sắc thể (NST), tính độ thích nghi và chọn lọc ra các NST thích hợp. Từ đó chọn một NST tốt nhất làm kết quả của thuật toán.

Biểu diễn các NST: Mỗi NST là một chuỗi các giá trị nhị phân diễn tả trạng thái hoạt động của luật tương ứng trong hệ SAM. Mỗi thế hệ sẽ chỉ sử dụng 10 NST. Thế hệ đầu tiên bao giờ cũng chứa đầy đủ các luật (tất cả giá trị bằng 1).

Hàm thích nghi: Mối quan hệ giữa kích thước SAM và độ chính xác trong xấp xỉ của SAM được giải quyết bằng hàm thích nghi (12):

- Các công thức:

$$Fit(m) = \ln \left(\frac{\sigma_\varepsilon^2}{\sigma_\varepsilon^2} \right) + \frac{\log_n(m)}{n}$$

m : Số luật (trạng thái 1) được sử dụng trong hệ SAM

n : Số bộ số liệu học.

$$\sigma_\varepsilon^2 = \frac{1}{n} \sum_{j=1}^n (y_j^d - F(x_j))^2$$

- **Mô tả thuật toán học tối ưu (HTU):**

b1. Khởi tạo 10 NST, có 1 NST biểu diễn đầy đủ các luật.

b2. Tính độ thích nghi Fit cho mỗi NST theo công thức (12).

b3. Chọn lại 5 NST có độ thích nghi cao nhất theo phương pháp chọn lọc xén.

b4. Nhân đôi quần thể để được 10 NST. Lại ghép nửa mặt nạ 10 NST chọn được với mặt nạ mới khởi tạo ngẫu nhiên.

b5. Nếu chưa gặp điều kiện dừng thì quay lại b2.

b6. Dừng thuật toán. Chọn một NST có độ thích nghi thỏa điều kiện dừng làm kết quả trả về.

Chuỗi nhị phân tìm được sẽ được dùng làm cơ sở cho việc hủy bỏ các luật không cần thiết trong hệ SAM.

4. Xây dựng ứng dụng

4.1. Bộ số liệu học

Lý thuyết hồi đáp [3] (Items Response Theory - còn gọi là Lý thuyết khảo thí hiện đại). Lý thuyết này bao gồm nhiều công cụ để đánh giá câu hỏi và đề thi trắc nghiệm. Trong bài báo này chúng tôi chỉ nghiên cứu công cụ đánh giá độ khó của câu hỏi trắc nghiệm.

Độ khó (P) của 1 câu hỏi trắc nghiệm là tổng số thí sinh trả lời đúng trên tổng số thí sinh dự thi. Có nhiều cách phân loại P, trong bài báo này P được phân loại như sau:

- $P < 30\%$: Câu hỏi rất khó
- $30\% \leq P < 40\%$: Câu hỏi khó
- $40\% \leq P < 60\%$: Câu hỏi khó vừa
- $60\% \leq P < 70\%$: Câu hỏi dễ
- $P \geq 70\%$: Câu hỏi rất dễ

Việc chọn ngưỡng để phân loại như trên là dựa theo phương pháp chuyên gia, được áp dụng trong [3]. Có thể áp dụng các ngưỡng khác thích hợp cho từng cơ sở giáo dục.

Từ đó, ta tính phần trăm số câu rất khó, khó, khó vừa của một đề thi. Cộng các phần trăm rất khó và khó để có cột (2) trong bảng 1, cột (3) trong bảng 1 là phần trăm số câu hỏi vừa.

Kết hợp với việc phân loại học lực của người học. Học lực ở đây được hiểu là kết quả học tập gần nhất của lớp X trước lúc thi đề thi Y nào đó. Ví dụ: nếu đề thi Y là đề thi kết thúc học phần học kỳ 2 thì ta sử dụng bảng phân loại học lực học kỳ 1 của lớp X. Cộng phần trăm số sinh viên có học lực khá, giỏi và xuất sắc để có cột (4) trong bảng 1. Cột (5) trong bảng 1 là phần trăm số sinh viên có học lực trung bình.

Lấy biểu điểm kết quả thi của lớp X sau khi thi xong với đề thi Y. Tính phần trăm số sinh viên có điểm xuất sắc, giỏi, khá và trung bình. Cộng phần trăm số sinh viên có điểm khá, giỏi, xuất sắc để có cột (6) trong bảng 1. Cột (7) trong bảng 1 là phần trăm số sinh viên có điểm thi trung bình.

Việc cộng các phần trăm như trên giúp giảm số chiều dữ liệu khi xử lý các ma trận U, V trong SAM.

Bảng 1. Bảng số liệu học

STT	Độ khó của đề thi (%)		Học lực của người học (%)		Biểu điểm (%)	
	(2)	(3)	(4)	(5)	(6)	(7)
	Khó	Vừa	KGX	TB	KGX	TB
1	16.29	17.11	54.22	17.99	44.44	45.12
2	43.03	31.61	51.12	47.63	41.68	10.13
...
Ntd	16.87	77.92	14.24	31.28	16.26	52.43

Trong đó:

- Ntd: Number of Training Data (Số lượng mẫu học).

- Khó: Rất khó + Khó.
- KGX: Khá + Giỏi + Xuất sắc.
- TB: Trung bình.

Nếu mẫu học thứ j được chọn đưa vào tập kiểm nghiệm (Test, xem 4.3.1) thì đầu vào của SAM là các cột (2), (3), (4), (5) và đầu ra mong muốn là cột (6) và (7). Như vậy, đầu ra của SAM là những số liệu tương tự như cột (6) và (7).

Mẫu học số 1 trong bảng 1 là đề thi tương đối dễ vì số lượng câu khó (rất khó + khó) và vừa chỉ chiếm 33.4 % còn lại là các câu hỏi dễ đến rất dễ, kết hợp với học lực của lớp sinh viên tương đối giỏi (số lượng sinh viên trung bình trở lên chiếm 72.21 %) nên biểu điểm của đề thi này các em đạt điểm từ trung bình trở lên chiếm 89.56 %.

4.2. Qui trình xây dựng ứng dụng

Từ việc ứng dụng SAM, kết hợp với bộ số liệu học như bảng 1, chúng tôi đưa ra qui trình dự báo gồm các bước như sau:

b1. Tập hợp xử lý số liệu học, tạo cơ sở dữ liệu theo bảng 1.

b2. Upload dữ liệu học và các thông số ban đầu.

b3. Xây dựng thủ tục gom cụm dữ liệu theo thuật toán (FCM).

b4. Xây dựng thủ tục tạo các tập mờ hình thang và các luật mờ.

b5. Xây dựng thủ tục tối ưu hệ luật theo thuật toán (HTU).

b6. Xây dựng thủ tục điều chỉnh thông số theo thuật toán (HTS).

b7. Xây dựng thủ tục dự báo theo công thức (3).

Chúng tôi đã tiến hành xây dựng ứng dụng theo qui trình trên, thực nghiệm chương trình được trình bày ở phần 4.3.

4.3. Thực nghiệm

Trường Cao đẳng Kinh tế Tài chính Vĩnh Long áp dụng hình thức thi trắc nghiệm khách quan trên máy tính cho một số học phần thông qua phần mềm Moodle. Khi kết thúc mỗi ca thi, chúng tôi kết xuất dữ liệu thô từ Moodle như bảng 2.

Bảng 2. Dữ liệu thô

Mã số Sinh viên	Thời gian thi	Tổng điểm	Câu 1	Câu 2	...	Câu n
...
0330020077	40 phút 49 giây	9	0	0.3		0.3
0330020078	42 phút 25 giây	8	0.3	0		0.3
...

Từ cột Câu 1 trong bảng 2 ta tính được độ khó của Câu 1 dựa vào số sinh viên trả lời đúng trên tổng số sinh viên dự thi. Số sinh viên trả lời không đúng có điểm số là 0. Tương tự cho các câu còn lại.

Khi đã tính độ khó của tất cả các câu trong đề thi, ta tính phần trăm số câu Rất Khó, Khó, Vừa ... Để có cột (2), (3) trong bảng 1.

Từ cột Tổng điểm trong bảng 2. Ta tính được phần trăm Khá, Giỏi, Xuất sắc và Trung bình để có cột (6), (7) trong bảng 1.

Dựa vào bảng phân loại học lực của số sinh viên này ở học kỳ trước ta có cột (4), (5) trong bảng 1.

4.3.1. Qui trình thực nghiệm

Số liệu thực nghiệm được chọn lọc từ các đề thi, học lực và kết quả thi của học phần Tin học đại cương thuộc các khóa lớp Cao đẳng chính quy từ năm 2005 đến nay tại Trường Cao đẳng Kinh tế Tài chính Vĩnh Long.

b1. Trong hơn 1000 mẫu học đã chọn lọc theo bảng 1. Chọn ra 100 mẫu làm tập thử nghiệm (Test), các mẫu còn lại đưa vào tập huấn luyện (Train).

b2. Huấn luyện SAM với tập Train ở b1.

b3. Dự báo với tập Test ở b1.

b4. Tính sai số của lần thử nghiệm hiện tại.

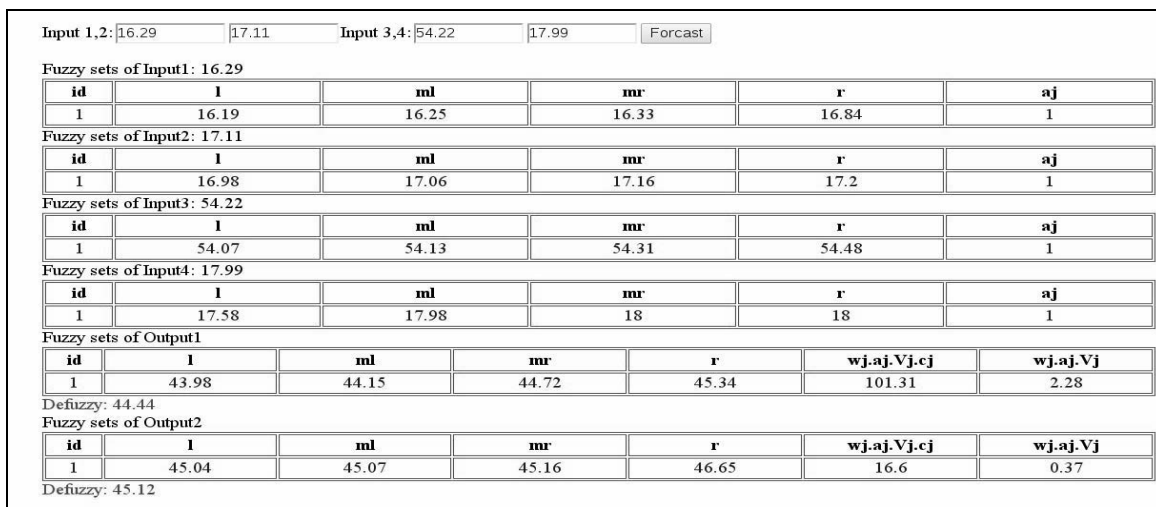
b5. Lặp lại các bước từ b1 đến b4 với 100 mẫu Test không trùng lặp với bước trước đó. Nếu các mẫu học đã được thử nghiệm hết thì đến b6.

b6. Tính trung bình cộng sai số của các lần thử nghiệm để được sai số cuối cùng.

4.3.2. Kết quả

Khi chạy ứng dụng này trên máy ASUS K40IJ có cấu hình CPU Intel Core 2 Duo T6670 tốc độ 2.2 GHz, RAM 4GB, Hệ điều hành Fedora 17 phiên bản 64 bit cho tổng thời gian từ lúc upload dữ liệu cho đến khi SAM sẵn sàng cho dự báo là 40 phút 23 giây. Trong đó thời gian học thông số chiếm nhiều nhất là 25 phút. Tiến hành đối sánh kết quả của SAM với kết quả mong muốn, chúng tôi đã:

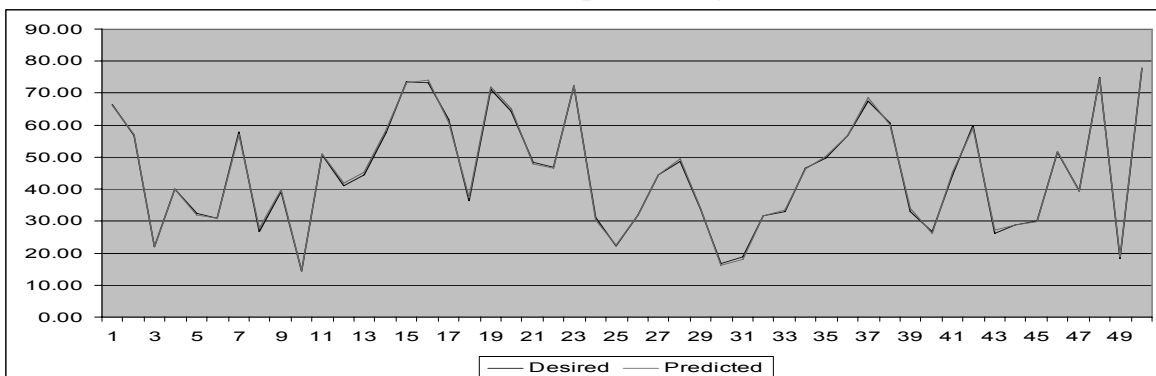
Cho lần lượt các mẫu thử trong tập Test (như bảng 1) vào để SAM dự báo. Nhập dữ liệu đầu vào là các cột (2), (3), (4), (5) tương ứng với input 1, 2, 3, 4 trong hình 2. Nhận được kết quả đối ứng với cột (6) và (7) như Defuzzy (khử mờ) trong hình 2.



Hình 2. Giao diện dự báo

Kết quả của quá trình thực nghiệm biểu của SAM với cột (6) của tập Test (kết quả diễn trong biểu đồ 1 chỉ đối sánh một đầu ra mong muốn).

Biểu đồ 1. Kết quả thực nghiệm



Trong đó:

- Trục ngang: Số thứ tự mẫu dữ liệu trong tập Test.
- Trục dọc: Cột (6) của mỗi mẫu dữ liệu trong tập Test.
- Đường màu xanh (kết quả mong muốn): Lấy số liệu gốc từ cột (6) tập Test.
- Đường màu hồng (kết quả dự báo): Lấy số liệu từ SAM tương ứng với tập Test.
- Sai số trung bình của hệ là ± 0.2 .

Trong biểu đồ 1 và trong hình 2, mẫu thử số 1 chính là mẫu thử có số thứ tự 1 trong bảng 1. Nghĩa là, nếu đề thi có cấu trúc độ khó 16.29% số câu rất khó + khó, 17.11% số câu khó vừa, đem cho một lớp sinh viên có học lực 54.22% em khá + giỏi + xuất sắc, 17.99% em trung bình thi, thì sẽ được biểu điểm 44.44% sinh viên đạt điểm khá + giỏi + xuất sắc và

45.12% sinh viên đạt điểm trung bình. Ở mẫu thử này, kết quả mong muốn và kết quả của SAM là trùng khớp với nhau nên đường màu hồng và màu xanh chồng lên nhau. Tại một số mẫu thử như số 8, 12, 13 .. đường màu xanh và màu hồng hơi chênh nhau một chút là do sai số của hệ tạo ra.

5. Đánh giá ứng dụng

5.1. Ưu điểm

Nhờ các cơ chế học trong SAM, ứng dụng cho kết quả dự báo khá khả quan. Qua 49 mẫu thử cho trong biểu đồ 1, SAM dự báo khá sát với thực tế (đường màu xanh và đường màu hồng gần khớp nhau).

5.2. Nhược điểm

Tuy SAM cho kết quả dự báo khá chính xác và chuyên gia huấn luyện SAM có thể tham gia vào quá trình điều chỉnh bên trong nó

nhưng khi số lượng mẫu học lớn và sai số mong muốn nhỏ thì thời gian học của SAM là vấn đề đáng quan tâm.

Kết quả dự báo là khá chính xác đối với hai thông số đầu vào là độ khó của đề thi và học lực của người học. Tuy nhiên, biểu điểm của một kỳ thi không chỉ do hai yếu tố này tạo nên mà còn rất nhiều yếu tố khác như: mục đích của kỳ thi, cơ sở vật chất, nhân lực phục vụ kỳ thi ...

6. Kết luận

Quá trình nghiên cứu và thực nghiệm chỉ ra rằng, có thể ứng dụng SAM vào việc dự báo

biểu điểm cho đề thi trắc nghiệm khách quan từ hai tham số độ khó của đề thi và học lực của người học. Từ đó phân nào giúp giảng viên ra đề, chọn đề thi trắc nghiệm sát với học lực người thi, tránh tình trạng ra đề thi quá khó hay quá dễ làm ảnh hưởng tới việc đánh giá kết quả học tập của người học. Hơn nữa, ứng dụng ra đời cũng góp phần nâng cao hiệu quả ứng dụng công nghệ thông tin và đổi mới trong công tác giáo dục và đào tạo, đặc biệt là lĩnh vực Khảo thí và Kiểm định chất lượng giáo dục, một lĩnh vực còn rất sơ khai trong việc ứng dụng các thành tựu khoa học máy tính.

Tài liệu tham khảo

- [1]. J. C. Bezdek (1981), "Pattern Recognition with Fuzzy Objective Function Algorithms", *Plenum Press, New York*;
- [2]. J. C. Dunn (1973), "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters", *Journal of Cybernetics 3: 32-57*;
- [3]. Nguyễn Thị Hạnh (2008), *Nghiên cứu thực trạng việc đánh giá kết quả học tập của sinh viên trường Cao đẳng Sư phạm trung ương*, Luận văn Thạc sỹ Quản lý giáo dục, Đại học Quốc gia Hà Nội;
- [4]. Dương Ngọc Hiếu, Võ Hoàng Tam, Nguyễn Thành Thi (2007), *Xây dựng thư viện mã nguồn mở cho bài toán dự báo*, Đề tài nghiên cứu khoa học cấp Trường, Trường Đại học Bách Khoa Thành phố Hồ Chí Minh;
- [5]. Hoàng Kiếm (2005), *Giải một bài toán trên máy tính như thế nào*, tập 2, NXB Giáo dục;
- [6]. Bart Kosko (1991), *Neural network and Fuzzy systems a dynamical systems approach to machine intelligence*, University of Southern California;
- [7]. Kim Ngân (2011), "Điểm thi môn Sử thấp: Do lỗi hệ thống hay do phương pháp giảng dạy", <http://dantri.com.vn/su-kien/diem-thi-mon-su-thap-do-loi-he-thong-hay-phuong-phap-giang-day-504319.htm>;
- [8]. Vũ Thanh Nguyên (2003), *Giải quyết một số vấn đề phân tích dự báo kinh tế ứng dụng trong ngành công nghiệp tại Thành phố Hồ Chí Minh*, Đề tài khoa học cấp Thành phố, Sở Khoa học và Công nghệ Thành phố Hồ Chí Minh;
- [9]. Phạm Thị Hoàng Nhung, Hà Quang Thụy (2007), "Nghiên cứu, sử dụng mạng nơ-ron nhân tạo trong dự báo lưu lượng nước đến hồ Hoà Bình trước 10 ngày", *Hội thảo Quốc gia Một số vấn đề chọn lọc về Công nghệ thông tin và Truyền thông*, lần thứ X, Đại Lải, Vĩnh Phúc.

APPLYING SCORE SPECTRUM PREDICTION TO OBJECTIVE TESTS BY STANDARD ADDITIVE MODEL

Summary

This paper proposes an application to predict the score spectrum of objective tests (OB) by Standard Additive Model (SAM). Nowadays the application of advanced technology in Testing & Evaluation (OT&E) is still limited. We have researched and successfully applied SAM to predict the score spectrum of OB. The results are positive.

Keywords: SAM, data prediction, objective test.

Ngày nhận bài: 26/8/2014; ngày nhận đăng: 18/12/2014.