DỰ ĐOÁN SỰ HÀI LÒNG VỀ CHẤT LƯỢNG DỊCH VỤ TƯỚI TIÊU TẠI ĐỒNG BẰNG SÔNG HỒNG DÙNG CÁC MÔ HÌNH HỒI QUY

Nguyễn Thanh Tùng[1]

[1] Khoa Công nghệ thông tin, Trường Đại học Thủy lợi

tungnt@tlu.edu.vn

TÓM TẮT *- Việc xác định mức độ hài lòng của người dân về dịch vụ tưới tiêu trong chính sách thủy lợi phí có ảnh hưởng lớn đến các tổ chức quản lý và khai thác công trình thuỷ lợi, ngân sách quốc gia và an sinh xã hội. Trong bài báo này, các mô hình hồi quy được áp dụng cho phân tích hồi quy đa biến nhằm mục đích dự đoán độ hài lòng của người dân về hệ thống tưới tiêu tại đồng bằng Sông Hồng. Kết quả thực nghiệm cho thấy mô hình hồi quy phi tuyến cho kết quả tốt hơn mô hình tuyến tính, tính đa dạng và khả thi của những mô hình dự đoán này có thể được áp dụng để xử lý các bài toán về kinh tế trong các lĩnh vực quản lý tài nguyên nước.*

Từ khóa *- Hồi quy đa biến, LASSO, k láng giềng, mạng nơron, véctơ hỗ trợ hồi quy, rừng ngẫu nhiên hồi quy, khai phá dữ liệu, máy học*

I. ĐẶT VẤN ĐỀ

Với mỗi hệ thống tưới tiêu cụ thể tại Việt Nam, việc đánh giá mức độ hài lòng của các hộ dùng nước tác động lớn đến chính sách thủy lợi phí của Chính phủ. Từ những nghiên cứu, phân tích định lượng liên quan đến sự hài lòng của người dân giúp Chính phủ điều chỉnh chính sách thủy lợi phí phù hợp nhằm nâng cao chất lượng dịch vụ tưới tiêu nông nghiệp. Trong nghiên cứu này, các mô hình hồi quy tiên tiến được nghiên cứu để phân tích, dự đoán mức độ hài lòng của người dân tại vùng đồng bằng Sông Hồng, từ đó lựa chọn mô hình phù hợp để áp dụng xử lý các bài toán về kinh tế, thủy văn trong thực tiễn.

Xét mô hình hồi quy tổng quát để giải bài toán xác định mức độ hài lòng của các hộ dân dùng dịch vụ nước tưới tiêu, thông thường được viết như sau:

$$\mathrm{Y} = \mathrm{f}(\mathrm{X}) + \epsilon, \tag{1}$$

trong đó ϵ là lỗi của mô hình, $\mathrm{E}(\epsilon) = 0, \mathrm{Var}(\epsilon) = \sigma_\epsilon^2$. Tập dữ liệu đầu vào $\mathcal{L} = (\mathrm{X_i}, \mathrm{Y_i})_{\mathrm{i}=1}^{\mathrm{N}}$ dùng để xây dựng mô hình hồi quy được thu thập, khảo sát độc lập từ các hộ dùng nước với các tiêu chí quan sát X (predictor features) và biến đích Y (response feature) lưu giá trị đánh giá mức độ hài lòng của các hộ dùng nước. Trong biểu thức (1), $\mathrm{X} \in \mathbb{R}^{\mathrm{M}}$ và $\mathrm{Y} \in \mathbb{R}^1$ là các biến ngẫu nhiên với xác suất $\mathcal{P}$, cụ thể, $\mathcal{P}(\mathrm{X} = \mathrm{x}, \mathrm{Y} = \mathrm{y})$ là xác suất mà các biến ngẫu nhiên X, Y nhận các giá trị x và y. Ở đây, M là số chiều của tập dữ liệu đầu vào và N là số mẫu thu thập được. Mục tiêu của bài toán hồi quy là tìm mô hình mà giá trị ước lượng của nó được dự đoán bởi hàm $\mathrm{f}(\cdot)$ có trung bình sai số bình phương (mean squared errors) càng nhỏ càng tốt. Các mô hình hồi quy trình bày trong bài báo này được dùng như 1 hàm $\mathrm{f}: \mathbb{R}^{\mathrm{M}} \to \mathbb{R}^1$ ước lượng giá trị $\mathrm{y} \in \mathrm{Y}$ tương ứng với dữ liệu đầu vào $\mathrm{x} \in \mathbb{R}^{\mathrm{M}}$.

Các nghiên cứu về đánh giá độ hài lòng của các hộ dùng nước tưới tiêu nói riêng và những bài toán kinh tế lượng nói chung ở Việt Nam, sau bước khảo sát và tiền xử lý số liệu, mô hình hồi quy tuyến tính thường được sử dụng để phân tích sự biến thiên của số liệu, dự báo mẫu trong tương lai. Mô hình tuyến tính được ưa dùng do dễ sử dụng, dễ cài đặt và việc diễn giải kết quả khá dễ hiểu. Tuy nhiên, kết quả hồi quy dùng mô hình tuyến tính thường có lỗi dự báo cao và gặp khó khăn khi dữ liệu phức tạp như có số liệu trống (missing value), số liệu không phải dạng số, số lượng biến gấp nhiều lần so với số lượng mẫu. Ngoài ra, lớp những mô hình tuyến tính cần những giả định như phân bố chuẩn, dữ liệu quan hệ tuyến tính để có được những kết quả dự báo hợp lý.

Trong nghiên cứu này, các mô hình hồi quy tuyến tính nhiều biến và phi tuyến được nghiên cứu áp dụng cho bài toán xác định mức độ hài lòng của các hộ dùng nước tưới tiêu tại đồng bằng Sông Hồng. Kỹ thuật kiểm tra chéo (k-folds cross validation) [10] được sử dụng cho các mô hình hồi quy trên tập huấn luyện để tìm tham số tối ưu dùng cho dự đoán dữ liệu kiểm thử. Độ đo sự quan trọng của các tiêu chí liên quan đến sự hài lòng của các hộ dùng nước tưới tiêu được phân tích, đánh giá và hiển thị trực quan giúp nhà quản lý có thêm thông tin cần thiết để đầu tư, nâng cấp dịch vụ tưới tiêu. Kết quả thực nghiệm trong bài báo này cho thấy mô hình phi tuyến cho kết quả dự đoán tốt hơn, đặc biệt là mô hình của tổ hợp các cây hồi quy, tính đa dạng của những mô hình hồi quy này có thể được ứng dụng giải quyết lớp các bài toán hồi quy trong lĩnh vực kinh tế ở Việt Nam.

II. CÁC MÔ HÌNH HỒI QUY

A. Mô hình hồi quy tuyến tính nhiều biến

Mô hình hồi quy tuyến tính gồm hồi quy đơn biến (single) và nhiều biến (multivariate). Hồi quy đơn biến là mô hình hồi quy với một biến hoặc đặc trưng (biến độc lập), hồi quy đa biến là mô hình hồi quy với nhiều biến và thường được sử dụng rộng rãi trong thực tế. Với tập dữ liệu đầu vào $\mathcal{L}$ cho trước, mô hình hồi quy tổng quát ở công thức (1) có thể được viết lại ở dạng sau [10]:

$$Y = E(Y|X) + \epsilon, \tag{2}$$

trong đó $\epsilon \sim N(0, \sigma^2)$ và

$$E(Y|X) = \beta_0 + \sum_{i=1}^{N} X_i \beta_i, \tag{3}$$

β_0 là hệ số chặn (intercept) và các β_i là độ dốc (slope). Để tìm các hệ số của mô hình, cách tiếp cận phổ biến là dựa trên phương pháp bình phương nhỏ nhất [11], trong đó chúng ta tìm các hệ số $\beta = (\beta_0, \beta_1, \dots, \beta_M)^T$ để cực tiểu hóa tổng bình phương phần dư (residual sum of squares, RSS):

$$RSS(\beta) = \sum_{i=1}^{N} \big(Y_i - E(Y|X)\big)^2 = \sum_{i=1}^{N} \left(Y_i - \beta_0 - \sum_{j=1}^{N} X_j \beta_j\right)^2. \tag{4}$$

Ta cần xác định véctơ β cho các hệ số trong mô hình hồi quy, giả thiết các điều kiện cho mô hình tuyến tính được đáp ứng (xem Huber [11]). Công thức (4) có thể được viết như sau: $RSS(\beta) = (Y - X\beta)^T (Y - X\beta)$. (5)

Nếu $X^T X$ không suy biến, véctơ β được xác định bằng phương trình sau:

$$\hat{\beta} = (X^T X)^{-1} X^T Y. \tag{6}$$

Từ (6) ta có phương trình hồi quy nhiều biến, để dự đoán giá trị mới $X = x$ ta tính đầu ra $\hat{Y}$ của mô hình hồi quy tuyến tính nhiều biến như sau:

$$\hat{Y} = X\hat{\beta} = (X^T X)^{-1} X^T Y. \tag{7}$$

Hồi quy LASSO

Phương pháp LASSO (Least absolute shrinkage and selection operator) [10], [18] là phương pháp hồi quy tuyến tính nhiều biến có hiệu chỉnh mô hình, phương pháp này đưa thêm hàm phạt vào hàm lỗi để lỗi hồi quy đạt nhỏ nhất:

$$RSS(\beta) = \sum_{i=1}^{N} \big(Y_i - E(Y|X)\big)^2 + \lambda \sum_{j=1}^{M} |\beta_j|. \tag{8}$$

Trong đó λ là hệ số phạt dùng để điều chỉnh mô hình, chuẩn L_1 được dùng cho việc dự đoán các tham số. Trong trường hợp λ đủ lớn sẽ có một số tham số hồi quy tiến dần về 0, do đó chúng không đóng vai trò gì trong mô hình hồi quy. Phương pháp LASSO cũng được dùng cho bài toán lựa chọn thuộc tính, với các biến có tham số hồi quy bằng 0 ta có thể loại khỏi mô hình.

B. Phương pháp hồi quy k láng giềng

Phương pháp k láng giềng dùng cho bài toán hồi quy không có quá trình huấn luyện để xây dựng mô hình học [10], khi dự đoán 1 mẫu mới, giải thuật tìm k (k=1, 2,..) láng giềng gần nhất của mẫu này trong tập dữ liệu huấn luyện $\mathcal{L}$, sau đó tính giá trị trung bình (hoặc trung vị) để trả về kết quả cuối cùng.

Quá trình tìm k láng giềng của mẫu mới thường sử dụng khoảng cách Euclidean được định nghĩa như sau:

$$\mathrm{d}(x_a, x_b) = \left(\sum_{j=0}^{M} (x_{aj} - x_{bj})^2\right)^{\frac{1}{2}}, \tag{9}$$

trong đó x_a và x_b là 2 mẫu độc lập.

C. Cây hồi quy

Mô hình cây hồi quy tách đệ quy theo hàng của tập dữ liệu đầu vào $\mathcal{L}$ thành các tập dữ liệu nhỏ hơn, hình thành nút và lá của cây. Tại mỗi lần tách nút, một thuộc tính và giá trị tách của thuộc tính này được chọn để chia nút thành 2 nút con, nút con trái và nút con phải.

1. Xây dựng cây hồi quy

Gọi t là nút cha để tách nhánh trên cây hồi quy. Việc tách nhánh trên thuộc tính X được xác định bởi việc giảm sự hỗn tạp [5] tại nút t, ký hiệu $\Delta R(X, t)$. Kỳ vọng của Y ở nút t được tối thiểu hóa nhờ hàm lỗi bình phương sai số được định nghĩa như sau:

$$R(t) = argmin_{Y_t \in \mathcal{L}}\, E[(Y_i - \overline{Y}_t)^2] = argmin_{Y_t \in \mathcal{L}}\, \frac{1}{N(t)} \sum_{X_i \in t} (Y_i - \overline{Y}_t)^2. \tag{10}$$

Trong đó $N(t)$ là tổng số mẫu hiện tại ở nút t và $\overline{Y}_t$ là trung bình mẫu của Y tại t.

Gọi s là giá trị chia tách thuộc tính X tại nút t thành nút con trái t_L và nút con phải t_R phụ thuộc vào $X \le s$ hoặc $X > s$, $t_L = \{X_i \in t, X_i \le s\}$ và $t_R = \{X_i \in t, X_i > s\}, i = 1..N$. Độ biến thiên của các mẫu cho mỗi nút con là

$$R(t_L) = \frac{1}{N_L(t)} \sum_{X_i \in t_L} \left(Y_i - \bar{Y}_{t_L}\right)^2, R(t_R) = \frac{1}{N_R(t)} \sum_{X_i \in t_R} \left(Y_i - \bar{Y}_{t_R}\right)^2. \quad (11)$$

Trong đó $\bar{Y}_{t_L}$ là trung bình mẫu của t_L và $N_L(t)$ là kích thước mẫu của t_L. Tương tự, $\bar{Y}_{t_R}$ và $N_R(t)$ là trung bình mẫu và kích thước mẫu của t_R.

Như vậy, việc giảm độ hỗn tạp theo việc chia tách s đối với X được tính như sau:

$$\Delta R(X,t) = R(t) - [R(t_L)p(t_L) + R(t_R)p(t_R)]. \quad (12)$$

Trong đó $p(t_L) = N_L(t)/N(t)$ và $p(t_R) = N_R(t)/N(t)$ là các tỷ lệ quan sát trong t_L và t_R. Điểm chia tách được chọn trên thuộc tính X cho mỗi nút t chính là giá trị làm cho $\Delta R(X,t)$ đạt cực đại.

2. Dự đoán dùng cây hồi quy

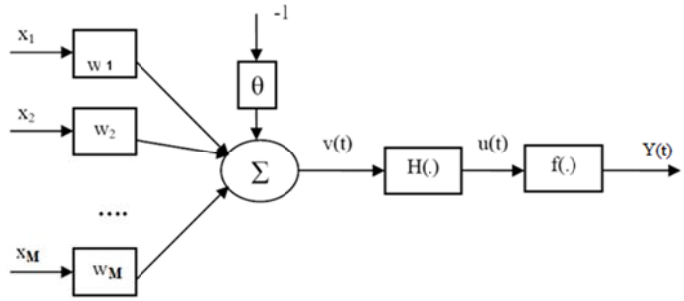
Khi xây dựng cây hồi quy, ta cần phải tính toán giá trị cho nút lá của cây, quá trình này được mô tả sau đây. Sử dụng các ký hiệu của Breiman [4], gọi θ là véctơ chứa tham số ngẫu nhiên để xác định việc xây dựng cây. Trong mỗi cây hồi quy, ta tính toán trọng số dương $w_i(x_i, \theta)$ cho mỗi mẫu $x_i \in \mathcal{L}$. Đặt $l(x, \theta, t)$ là nút lá t trong cây hồi quy. Các mẫu $x_i \in l(x, \theta, t)$ được gán các trọng số $w_i(x, \theta) = 1/N(t)$, trong đó N_t là số mẫu trong $l(x, \theta, t)$. Nghĩa là việc dự đoán dùng cây hồi quy đơn giản là tính giá trị trung bình của các mẫu tại nút lá của cây.

Với dữ liệu thử nghiệm $X = x$, $\hat{Y}$ là giá trị dự đoán của cây hồi quy được tính như sau:

$$\hat{Y} = \sum_{i=1}^{N} w_i(x, \theta) Y_i = \sum_{x_i, X_i \in l(x,\theta,t)} w_i(x, \theta) Y_i. \quad (13)$$

D. Mạng nơron nhân tạo

Mạng nơron nhân tạo giả lập quá trình học tập và tính toán của bộ não con người [1], [16]. Một mạng nơron nhân tạo được xây dựng từ những thành phần cơ sở là những nơron nhân tạo gồm nhiều đầu vào và một đầu ra (Hình 1). Mỗi nơron nhân tạo giả lập một nơron sinh học, gồm một ngưỡng kích hoạt (bias) và một hàm kích hoạt (hay hàm truyền –transfer function), đặc trưng cho tính chất của nơron. Các nơron nhân tạo được liên kết với nhau bằng các kết nối. Mỗi kết nối có trọng số kết nối (weight), đặc trưng cho khả năng nhớ của mạng nơron. Quá trình huấn luyện mạng nơron là 1 quá trình điều chỉnh các ngưỡng kích hoạt và các trọng số kết nối, dựa trên dữ liệu học.



Hình 1. Kiến trúc một nơron nhân tạo

Trong đó:

v(t): Tổng tất cả các đầu vào mô tả toàn bộ thế năng tác động ở thân nơron.

$X_k(t)$: Các biến đầu vào (các đặc trưng), k=1..M.

w_k: Trọng số liên kết ngoài giữa các đầu vào k với nơron hiện tại.

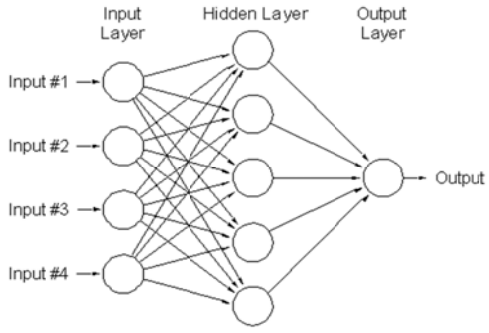
H(.): Hàm kích hoạt.

Y(t): Tín hiệu đầu ra nơron.

θ: Ngưỡng (là hằng số), xác định ngưỡng kích hoạt.

Khi kết hợp các nơron lại với nhau ta có một mạng nơron nhân tạo. Tuỳ theo cách thức liên kết giữa các nơron mà ta có các loại mạng khác nhau như: mạng truyền thẳng (Hình 2), mạng phản hồi,… Ta có thể xem như mạng nơron nhân tạo biểu thị mô hình hồi quy theo công thức (1) với X là véctơ số liệu đầu vào và Y là véctơ số liệu đầu ra. Ưu điểm của một mạng nơron nhân tạo là nó cho phép xây dựng một mô hình tính toán có khả năng học dữ liệu rất cao. Có

thể coi mạng nơron nhân tạo là một hộp đen có nhiều đầu vào và nhiều đầu ra có khả năng học được mối quan hệ giữa đầu ra và đầu vào dựa trên dữ liệu được học.



Hình 2. Mạng nơron lan truyền thẳng

Quá trình huấn luyện mạng nơron dựa trên lỗi hồi quy giữa giá trị dự đoán và giá trị quan sát được của biến đích, giải thuật huấn luyện sẽ điều chỉnh các trọng số kết nối của mạng nơron nhằm cực tiểu hóa lỗi hồi quy trên các mẫu huấn luyện. Sau khi mạng được huấn luyện thành công, các tri thức tích luỹ được trong quá trình huấn luyện mạng (các ma trận trọng số, các tham số tự do, v.v) sẽ được cập nhật vào cơ sở tri thức để sử dụng trong quá trình dự đoán. Có nhiều loại mạng nơron, nhiều tầng và được dùng cho cả bài toán học có giám sát và học không giám sát. Trong nghiên cứu này, chúng tôi cài đặt mạng nơron 1 lớp truyền thẳng, sử dụng trọng số suy giảm (weight decay) và hệ số co của mô hình để λ tránh tình trạng học vẹt (over-fitting), xem thêm ở [16].

E. Máy véctơ hỗ trợ hồi quy

Máy véctơ hỗ trợ hồi quy (Support Vector Regression, SVR) [17] tìm siêu phẳng đi qua tất cả các điểm dữ liệu với độ lệch chuẩn ε. Trong hồi quy ε – SV, mục đích là tìm một hàm f(X) trong công thức (1) có sai số nhỏ nhất ε so với biến đích Y_i:

$$f(X) = w^T\Phi(X) + b, \tag{14}$$

Trong đó $w \subset R^M$, $\Phi(X)$ biểu thị một hàm phi tuyến được chuyển từ không gian R^M vào không gian nhiều chiều. Mục đích ở đây là cần tìm w và b để giá trị X=x có thể được xác định bằng cách tối thiểu hóa lỗi hồi quy. Từ đó dẫn đến giải bài toán quy hoạch toàn phương như sau:

$$\min \Phi(w, b, \xi, \xi^*) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{15}$$

Với điều kiện:

$$\begin{cases} Y_i - (wX_i + b) \le \varepsilon + \xi_i \\ (wX_i + b) - Y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases}$$

Ở đây, ξ_i, ξ_i^* là hai biến bù [17] và C > 0 dùng để chỉnh độ rộng giữa lề và lỗi. Để giải quyết bài toán (15), trước tiên phải tìm cực tiểu của hàm L theo w, b, ξ_i, ξ_i^*.

$$\min L(w, b, \alpha, \alpha^*, \xi, \xi^*, \eta, \eta^*) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) - \sum_{i=1}^{N}(\eta_i\xi_i + \eta_i^*\xi_i^*)$$
$$- \sum_{i=1}^{N}\alpha_i(\varepsilon + \xi_i + Y_i - w^T\Phi(X_i) - b) - \sum_{i=1}^{N}\alpha_i^*(\varepsilon + \xi_i^* - Y_i + w^T\Phi(X_i) + b). \tag{16}$$

Với η_i, η_i^*, α_i, α_i^* là các hệ số Lagrange và thỏa mãn điều kiện: $\eta_i, \eta_i^*, \alpha_i, \alpha_i^* \ge 0$, i=1..N.

Lấy đạo hàm cấp 1 của phương trình (16), hồi quy phi tuyến SVR sử dụng hàm lỗi ε được tính như sau:

$$max\left\{-\frac{1}{2}\sum_{1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\Phi(X_i, X_j) - \varepsilon\sum_{1}^{N}(\alpha_i + \alpha_i^*) + \sum_{1}^{N}Y_i(\alpha_i - \alpha_i^*)\right\} \tag{17}$$

với ràng buộc:

$$\sum_1^N(\alpha_i - \alpha_i^*) = 0;\ \alpha_i, \alpha_i^* \in [0, C]. \tag{18}$$

Giải biểu thức (17) với ràng buộc (18) xác định được các nhân tử Lagrange α_i, α_i^*. Khi đó, mô hình hồi quy SVR được trình bày ở (14), với

$$\widehat{w} = \sum_{1}^{N}(\alpha_i - \alpha_i^*)X_i, \quad \hat{b} = -\frac{1}{2}\widehat{w}\left(X_j + X_k\right).$$

Trong đó X_j và X_k là 2 véctơ hỗ trợ, $\alpha_i \in (0, C)$ và $\alpha_i^* \in (0, C)$.

SVR có thể dùng các hàm nhân khác nhau để giải quyết lớp các bài toán hồi quy phi tuyến mà không cần bất kỳ một thay đổi nào về mặt thuật toán, các hàm nhân được dùng thông dụng như:

- Hàm nhân đa năng Gaussian RBF có dạng: $K(u, v) = e^{-\sigma||u-v||^2}$.
- Hàm nhân đa thức bậc d > 0: $K(u, v) = (C + u.v)^d$.

F. Rừng ngẫu nhiên hồi quy

Rừng ngẫu nhiên hồi quy (RF) [3], [4] gồm tập hợp các cây hồi quy đã trình bày ở mục ***II. C***. Từ tập dữ liệu đầu vào $\mathcal{L}$, RF dùng kỹ thuật lấy mẫu bootstrap có hoàn lại tạo ra nhiều tập dữ liệu khác nhau. Trên mỗi tập dữ liệu con này, lấy ngẫu nhiên một lượng cố định thuộc tính, thường gọi là *mtry* để xây dựng cây. Mỗi cây hồi quy được xây dựng không cắt nhánh với chiều cao tối đa. Việc lấy hai lần ngẫu nhiên cả mẫu và thuộc tính đã tạo ra các tập dữ liệu con khác nhau giúp RF giảm độ dao động (variance) của mô hình học.

1. Dự đoán bằng rừng ngẫu nhiên hồi quy

Việc xây dựng rừng ngẫu nhiên hồi quy và dự đoán mẫu mới được mô tả như sau. Đặt $\Theta = \{\theta_k\}_1^K$ là tập gồm K các véctơ tham số ngẫu nhiên cho rừng được sinh ra từ $\mathcal{L}$, trong đó θ_k là một véctơ tham số ngẫu nhiên để xác định độ lớn của cây thứ k trong rừng (k = 1..K). Gọi $\mathcal{L}_k$ là tập dữ liệu thứ k sinh ra từ $\mathcal{L}$ dùng kỹ thuật bootstrap, trong mỗi cây hồi quy T_k từ $\mathcal{L}_k$, ta tính trọng số dương $w_i(x_i, \theta_k)$ cho từng mẫu $x_i \in \mathcal{L}$. Đặt $l(x, \theta_k, t)$ là nút lá t trong cây T_k. Mẫu $x_i \in l(x, \theta_k, t)$ được gán cùng một trọng số $w_i(x, \theta_k) = 1/N(t)$, trong đó $N(t)$ là số các mẫu trong $l(x, \theta_k, t)$. Trong trường hợp này, tất cả các mẫu trong $\mathcal{L}_k$ được gán trọng số dương và các mẫu không trong $\mathcal{L}_k$ được gán bằng 0.

Với một cây hồi quy T_k, khi có giá trị thử nghiệm $X = x$ thì giá trị dự đoán $\hat{Y}_k$ tương ứng:

$$\hat{Y}_k = \sum_{i=1}^{N} w_i(x, \theta_k)Y_i = \sum_{x_i, X_i \in l(x,\theta_k,t)} w_i(x, \theta_k)Y_i. \quad (18)$$

Trọng số $w_i(x)$ được tính bởi rừng ngẫu nhiên là giá trị trung bình của các trọng số dự đoán của tất cả các cây trong rừng. Công thức tính như sau:

$$w_i(x) = \frac{1}{K}\sum_{k=1}^{K} w_i(x, \theta_k). \quad (19)$$

Cuối cùng, giá trị dự đoán của rừng ngẫu nhiên hồi quy được cho bởi:

$$\hat{Y} = \sum_{i=1}^{N} w_i(x)Y_i. \quad (20)$$

2. Độ đo sự quan trọng của thuộc tính

Khi cây hồi quy phân chia tập dữ liệu đầu vào thành các vùng không giao nhau (theo hàng), giá trị dự đoán là giá trị trung bình được gán vào các vùng tương ứng (lá của cây). Tại mỗi bước tính toán để tách nút t, theo công thức (12) tất cả các giá trị của mỗi thuộc tính X được xét để tìm điểm tách khi đạt độ giảm hỗn tạp (impurity) $\Delta R(X, t)$ là lớn nhất. Do đó, trong quá trình xây dựng cây hồi quy, việc giảm sự hỗn tạp trên từng thuộc tính cụ thể được dùng để tính độ đo sự quan trọng của thuộc tính khi dùng mô hình cây [5].

Với mô hình rừng ngẫu nhiên, độ đo sự quan trọng của thuộc tính X được tính bằng cách lấy giá trị trung bình của tất cả các độ đo của các cây hồi quy độc lập. Có một điểm lợi trong việc tính độ đo sự quan trọng của thuộc tính dùng mô hình rừng ngẫu nhiên là độ đo của các biến có tương tác lẫn nhau đều được xem xét một cách tự động, điều này khác hẳn với những phương pháp tính tương quan tuyến tính như Kendall, Pearson. Độ đo sự quan trọng của thuộc tính X còn được tính theo cách khác dùng phương pháp lặp hoán vị [13], [14] cho kết quả chính xác hơn, tuy nhiên thời gian tính toán lâu hơn do chạy nhiều lần rừng ngẫu nhiên trên tập dữ liệu mở rộng cỡ 2M chứa các biến giả.

Gọi $IS_k(X_j)$, IS_{X_j} lần lượt là độ đo sự quan trọng của thuộc tính X_j trong một cây hồi quy T_k(k=1...K) và trong một rừng ngẫu nhiên. Từ công thức (12), ta tính độ đo sự quan trọng của X_j từ cây hồi quy độc lập như sau:

$$IS_k(X_j) = \sum_{t \in T_k} \Delta R(X_j, t), \quad (20)$$

và từ rừng ngẫu nhiên là:

$$IS_{X_j} = \frac{1}{K}\sum_{k=1}^{K} IS_k\left(X_j\right). \quad (21)$$

G. Boosting

Mô hình boosting [6], [7] ban đầu được phát triển xử lý bài toán phân lớp sau đó được mở rộng cho bài toán hồi quy. Trong mục này, kỹ thuật điển hình của boosting là AdaBoost (*Adaptive Boost*) được trình bày vắn tắt, sau đó mô hình boosting của Friedman với hàm cơ sở là cây hồi quy được áp dụng xử lý bài toán dự đoán sự hài lòng của các hộ dân dùng nước tưới tiêu.

Adaboost là một bộ phân loại mạnh phi tuyến dựa trên hướng tiếp cận boosting được Freund và Schapire đưa ra vào năm 1996 xử lý bài toán phân lớp nhị phân [8]. Adaboost hoạt động trên nguyên tắc kết hợp tuyến tính các phân loại yếu để hình thành một phân loại mạnh. Để có thể kết hợp các bộ phân loại yếu, adaboost sử dụng một trọng số (weight) để đánh dấu các mẫu khó nhận dạng. Trong quá trình huấn luyện, cứ mỗi phân loại yếu được xây dựng, thuật toán sẽ tiến hành cập nhật lại trọng số để chuẩn bị cho việc xây dựng phân loại yếu tiếp theo: tăng trọng số của các mẫu bị nhận dạng sai và giảm trọng số của các mẫu được nhận dạng đúng bởi phân loại yếu vừa xây dựng. Bằng cách này, các phân loại yếu sau có thể tập trung vào các mẫu mà các phân loại yếu trước đó chưa thực hiện tốt. Sau cùng các phân loại yếu sẽ được kết hợp tùy theo mức độ 'tốt' của chúng để tạo nên một phân loại mạnh.

Các bước thực hiện thuật toán AdaBoost như sau:

- Khởi tạo trọng số ban đầu cho tất cả các mẫu: với *m* là số mẫu đúng (ứng với các mẫu có nhãn Y = 1) và *l* là số mẫu sai (có nhãn tương ứng Y = -1).

$$w_{1,k} = \frac{1}{2m}, \frac{1}{2l} \tag{22}$$

- Xây dựng T các phân loại yếu. Lặp t = 1, …, T.
  - Với mỗi mẫu trong $\mathcal{L}$, xây dựng một phân loại yếu h_j với ngưỡng θ_j và lỗi ε_j.

$$\varepsilon_j = \sum_{k=1}^{N} w_{t,k} \left| h_j(X_k) - Y_k \right| \tag{23}$$

  - Chọn ra h_j với ε_j nhỏ nhất, ta được $h_t: X \to \{1, -1\}$
  - Cập nhật lại trọng số:

$$w_{t+1,k} = \frac{w_{t,k}}{Z_t} \times \begin{cases} e^{-\alpha_t}, & h_t(x_k) = y_k \\ e^{\alpha^t}, & h_t(x_k) \neq y_k \end{cases} \tag{24}$$

  Trong đó: $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_j}{\varepsilon_j}\right)$, (25)

  và hệ số Z_t dùng để đưa w_{t+1} về đoạn [0,1] (normalization factor).
- Phân loại mạnh được xây dựng :

$$H(x) = dấu\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right). \tag{26}$$

Friedman [9] đề xuất mô hình máy boosting dùng hàm học cơ sở là cây quyết định xử lý được cả bài toán phân lớp và hồi quy. Ý tưởng chính khi xây dựng mô hình hồi quy như sau: Mô hình học ban đầu khởi tạo với cây hồi quy và hàm lỗi cho trước (thường dùng hàm lỗi bình phương), giải thuật tìm mô hình cực tiểu hóa lỗi hồi quy. Bước đầu tiên, giải thuật dự đoán biến đầu ra $\hat{Y}_i$ bằng cách lấy giá trị trung bình các biến quan sát được Y_i. Tiếp theo lặp lại K lần (số cây hồi quy K là tham số của mô hình) để thực hiện: (i) Tính toán phần dư $\tilde{E} = Y_i - \hat{Y}_i$ và xây dựng mô hình cây hồi quy dùng phần dư $\tilde{E}$ là biến đích với mục tiêu cực tiểu hóa lỗi. (ii) Dự đoán mẫu dùng mô hình cây hồi quy ở bước trước đó. (iii) Cập nhật $\hat{Y}_i$ bằng cách thêm các giá trị dự đoán ở lần lặp trước vào các giá trị dự đoán được tạo ra trong bước trước đó. Mô hình Boosting dùng cây hồi quy khác rừng ngẫu nhiên khi các cây trong Boosting có đóng góp khác nhau khi đưa ra kết quả dự đoán cuối và cây hồi quy sau được xây dựng phụ thuộc cây trước, ngoài ra chúng được xây dựng với chiều cao biết trước còn ở rừng ngẫu nhiên các cây hồi quy được xây dựng độc lập và không cắt nhánh.

III. KẾT QUẢ THỰC NGHIỆM

A. Mô tả dữ liệu

Dữ liệu dùng trong thực nghiệm được thu thập tại vùng đồng bằng Sông Hồng (tỉnh Thái Bình, Nam Định, Bắc Ninh và Hà Nội) gồm 480 hộ dùng nước (mẫu quan sát) và 05 nhóm tiêu chí sau[1]:

- Tính hữu hình (Tangibility) gồm 7 biến quan sát:
  - Các hệ thống tưới, tiêu có chất lượng tốt, đảm bảo chuyển nước và phân phối nước đến các diện tích cần tưới, tiêu (HH1).
  - Các đơn vị cung cấp dành đủ kinh phí cho công tác quản lý, vận hành và bảo dưỡng hệ thống tưới, tiêu (HH2).
  - Nhân viên thủy lợi mặc đồng phục đơn vị (HH3).
  - Tổ chức cung cấp nước có tài liệu hướng dẫn quản lý vận hành công trình thủy lợi (HH4).
  - Hợp đồng cung cấp dịch vụ được trình bày rất dễ hiểu (HH5).
  - Các thiết bị của tổ chức cung cấp nước có chất lượng tốt (HH6).

[1] Phần trong ngoặc viết tắt tên biến dùng cho huấn luyện mô hình hồi quy

- Việc duy tu, bảo dưỡng hệ thống tưới được thực hiện đều đặn và khi cần (HH7).

- Độ tin cậy (Reliability) gồm 4 biến quan sát:
  - Đơn vị cung cấp dịch vụ tưới, tiêu giới thiệu đầy đủ nội dung hợp đồng với tổ chức cung cấp nước cũng như các kỹ thuật và cách sử dụng khi ông bà muốn đăng ký sử dụng (STC1).
  - Tổ chức cung cấp nước thực hiện đúng dịch vụ tưới tiêu như hợp đồng (STC2)
  - Tổ chức cung cấp nước xử lý sự cố ngay khi công trình hư hỏng, xuống cấp (STC3).
  - Từ năm 2008 đến nay tổ chức cung cấp nước không để xảy ra bất kỳ sai sót nào khi tính chi phí hàng tháng (STC4)
- Độ đáp ứng (Responsiveness) gồm 9 biến quan sát
  - Nhân viên thủy lợi cho ông bà biết khi nào thực hiện dịch vụ tưới tiêu (DDU1).
  - Nhân viên thủy lợi nhanh chóng thực hiện dịch vụ cho ông bà (DDU2).
  - Tổ chức cung cấp nước thực hiện đúng lịch cấp nước (DDU3).
  - Tổ chức cung cấp nước cung cấp tối đa khả năng cấp nước (DDU4).
  - Khối lượng nước cấp đáp ứng tốt nhu cầu theo từng giai đoạn sinh trưởng, phát triển của cây trồng (DDU5).
  - Nhân viên thủy lợi cung cấp luôn luôn sẵn sàng đáp ứng yêu cầu của ông bà (DDU6).
  - Chất lượng nước tưới được đảm bảo (DDU7).
  - Thời gian khắc phục hư hỏng nhanh chóng (DDU8).
  - Ông bà không bao giờ phải lặp lại các khiếu nại trước (DDU9).
- Sự đảm bảo (Assurance) gồm 7 biến quan sát:
  - Cách cư xử của nhân viên gây niềm tin cho ông bà (SBD1).
  - Ông bà cảm thấy rất an toàn khi giao dịch với tổ chức cung cấp nước (SBD2).
  - Nhân viên thủy lợi có đủ hiểu biết để trả lời tất cả các câu hỏi của ông bà liên quan đến hệ thống tưới, tiêu (SBD3).
  - Nhân viên thủy lợi của tổ chức cung cấp nước luôn luôn niềm nở với ông bà (SBD4).
  - Thời gian phân phối nước tới các thửa ruộng luôn luôn đủ nước trong mỗi đợt tưới (SBD5).
  - Từ năm 2008 đến nay nhân viên thủy lợi trả lời được tất cả các thắc mắc của ông bà liên quan đến số tiền ông bà trả trong tháng (SBD6).
  - Nhân viên thủy lợi rất nhanh khắc phục khi hệ thống tưới, tiêu có sự cố (SBD7).
- Sự đồng cảm (Empathy) gồm 7 biến quan sát:
  - Nhân viên kỹ thuật thủy lợi luôn làm việc vào những giờ thuận tiện cho ông bà (SDC1).
  - Không có bất cứ ai ở Tổ chức cung cấp nước quan tâm đến những bức xúc của ông bà về dịch vụ tưới, tiêu (SDC2).
  - Lịch phân phối nước rất thuận tiện theo giờ sản xuất của gia đình ông bà (SDC3).
  - Ông bà được quan tâm và chú ý mỗi khi thắc mắc về dịch vụ tưới, tiêu (SDC4).
  - Tổ chức cung cấp nước điều chỉnh lịch tưới phù hợp với sự thay đổi của thời tiết (SDC5).
  - Nhân viên của tổ chức cung cấp nước luôn hiểu rõ những nhu cầu của ông bà (SDC6).
  - Đơn vị cung cấp lấy lợi ích của ông bà là mục tiêu phát triển bền vững của họ (SDC7).

Biến đích đo sự hài lòng (SHL) của các hộ dùng nước có giá trị kiểu thập phân, SHL $\in$ [0.0, 10.0], giá trị càng cao càng phản ánh sự hài lòng về chất lượng dịch vụ tưới tiêu. Các tiêu chí đo lường chất lượng dịch vụ ở trên được lấy theo mô hình Servqual do Parasuraman và đồng nghiệp [15] đề xuất, phương pháp Cronbach Alpha [2] cũng được dùng để kiểm định độ tin cậy của các biến, tiền xử lý chúng trước khi đưa vào các mô hình hồi quy để huấn luyện.

B. Tham số mô hình và phương pháp đánh giá

Chúng tôi dùng căn bình phương sai số (Root mean squared error-RMSE), sai số tuyệt đối (mean absolute error-*MAE*) và hệ số xác định bội (coefficient of determination) R^2 để đánh giá tính hiệu quả của các mô hình hồi quy:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2};\ MAE = \frac{1}{N}\sum_{i=1}^{N}|Y_i - \hat{Y}_i| \text{ và } R^2 = 1 - \sum_{i=1}^{N}(Y_i - \hat{Y}_i)/\sum_{i=1}^{N}(Y_i - \bar{Y}_i).$$

Trong đó: Y_i, $\hat{Y}_i$ và $\bar{Y}$ chỉ giá trị thực, giá trị dự đoán và giá trị trung bình của mẫu thứ i tương ứng. Mô hình hồi quy cho kết quả tốt là mô hình đạt được sai số RMSE và MAE nhỏ. Giá trị R^2 cao là một dấu hiệu cho thấy mối liên hệ giữa các tiêu chí và biến số SHL chặt chẽ. Giá trị R^2 càng cao cho thấy mô hình sử dụng để phân tích có khả năng giải thích càng tốt các khác biệt về độ hài lòng giữa các hộ dùng nước.

Gói phần mềm *caret* [12] được sử dụng để tiến hành các thực nghiệm trên môi trường R, các mô hình hồi quy liệt kê ở mục II đều được tích hợp trong gói phần mềm này. Từ tập dữ liệu ban đầu gồm 480 mẫu quan sát và 34 tiêu chí, chúng tôi chia làm 2 phần để huấn luyện và kiểm thử mô hình, tập dữ liệu huấn luyện gồm 336 mẫu (70%) và tập dữ liệu kiểm thử gồm 144 mẫu (30%). Khi xây dựng mô hình hồi quy, chúng tôi sử dụng kỹ thuật kiểm tra chéo 5-folds với 2 lần lặp và dựa trên hàm lỗi RMSE để tìm tham số tối ưu của từng mô hình, sau đó lựa chọn mô hình có RMSE nhỏ nhất với tham số tìm được để dự đoán dữ liệu kiểm thử. Kỹ thuật kiểm tra chéo cũng cho phép tính hệ số xác định bội R^2 phản ánh khả năng giải thích của từng mô hình hồi quy. Các thực nghiệm được tiến hành trên 2 máy phục vụ dùng hệ điều hành Windows Server 2012 64-bit, mỗi máy có cấu hình IntelR XeonR CPU E5-2640 2.5 GHz, 24 cores, 8 MB cache và 128 GB RAM. Các mô hình đều được cài đặt song song sử dụng hết 24 cores trên mỗi máy để huấn luyện, tìm tham số tối ưu và các thực nghiệm khác.

C. Kết quả dự đoán độ hài lòng về chất lượng dịch vụ tưới tiêu

Kết quả các mô hình hồi quy dự đoán độ hài lòng của các hộ dùng nước về dịch vụ tưới tiêu được trình bày trong Bảng 1. Ở 3 cột R^2, RMSE và MAE kết quả dự đoán với R^2 cao nhất và lỗi dự đoán thấp nhất được in đậm và gạch dưới, các kết quả tốt thứ nhì và thứ ba được in với số lượng dấu (**) và (***) tương ứng.

Ta có thể dễ dàng nhận thấy mô hình hồi quy tuyến tính nhiều biến có kết quả dự đoán kém nhất, mô hình LASSO có cải thiện khả năng dự đoán hơn so với mô hình tuyến tính nhiều biến nhưng kết quả kiểm thử vẫn kém xa các mô hình khác. Các mô hình hồi quy phi tuyến tỏ rõ ưu thế hơn, cụ thể như rừng ngẫu nhiên, mạng nơ-ron nhân tạo và k láng giềng có kết quả dự đoán với lỗi hồi quy nhỏ. Mô hình cây hồi quy cho kết quả kém nhất theo R^2 và RMSE, mô hình máy véctơ hỗ trợ hồi quy và mô hình boosting có kết quả dự đoán chỉ hơn mô hình tuyến tính trên tập dữ liệu kiểm thử đang tiến hành thực nghiệm. Kết quả trình bày ở Bảng 1 cũng cho thấy mô hình k láng giềng đạt lỗi MAE thấp nhất, đây là phương pháp hồi quy phi tuyến khá hiệu quả, mô hình có khả năng dự đoán đạt độ chính xác cao trong khi thời gian tính toán nhanh. Tuy nhiên, xét khả năng dự đoán của các mô hình hồi quy liệt kê tại Bảng 1, ta có thể thấy rõ mô hình rừng ngẫu nhiên dự đoán chính xác nhất.

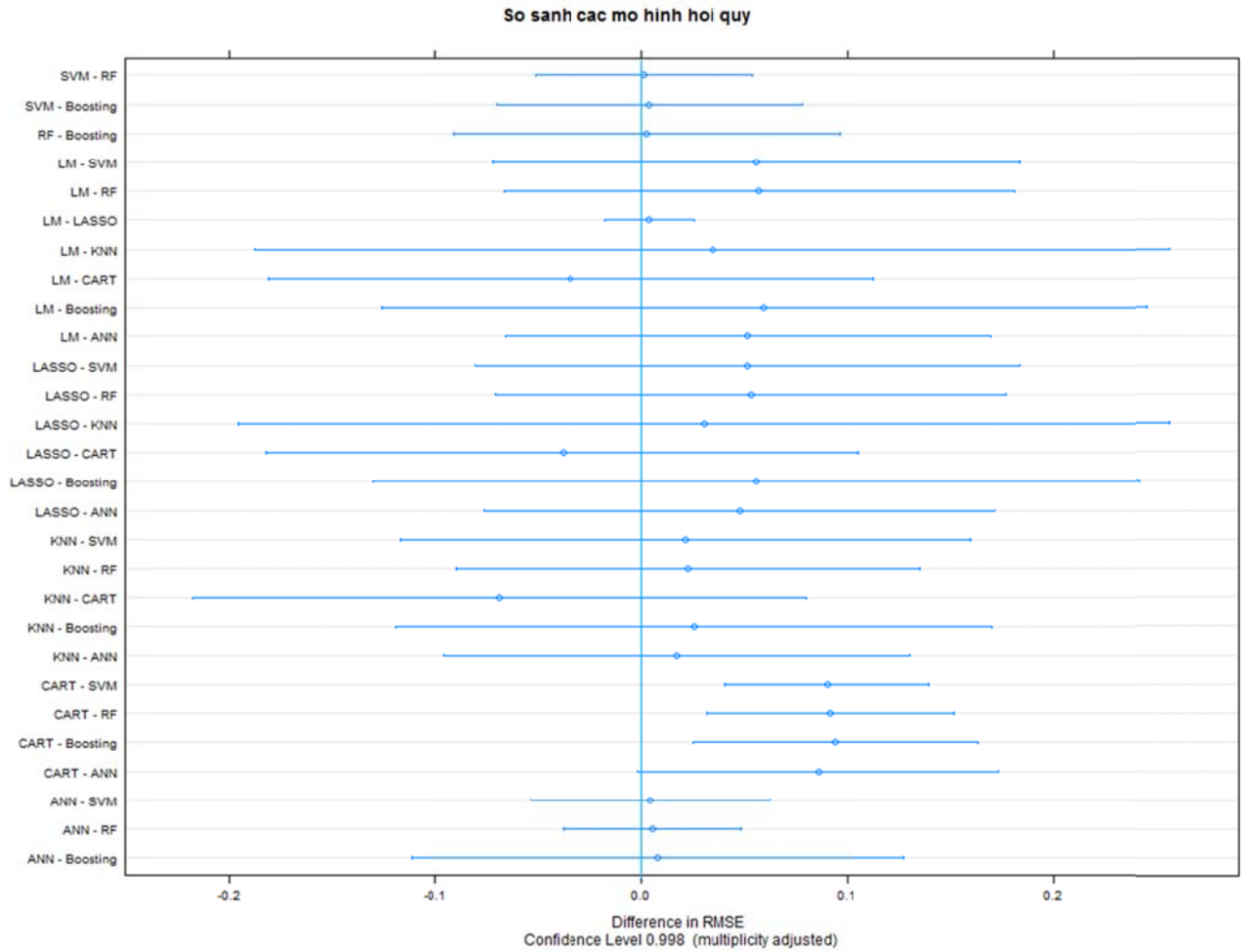
Bảng 1. Kết quả của các mô hình hồi quy dự đoán độ hài lòng về chất lượng dịch vụ tưới tiêu trên dữ liệu kiểm thử.

TT	Mô hình hồi quy	Tham số tối ưu	R^2	RMSE	MAE
1	Hồi quy tuyến tính (LM)	Mặc định	0.839	0.267	0.167
2	Hồi quy LASSO	λ= 0.01	0.844	0.263	0.163
3	K láng giềng (KNN)	k = 1	**0.894	**0.216	**<u>0.085</u>**
4	Cây hồi quy (CART)	Complexity parameter (cp)=0	0.835	0.272	0.156
5	Mạng nơ ron nhân tạo (ANN)	Trọng số phân rã=0.1 và số nơ-ron=9	***0.892	***0.218	**0.106
6	Máy véctơ hỗ trợ (SVR)	RBF, σ = 0.032, ε=0.1 và C = 32	0.852	0.255	0.143
7	Rừng ngẫu nhiên (RF)	mtry = 9 và K=1000	**<u>0.902</u>**	**<u>0.208</u>**	***0.107
8	Boosting	K = 500, interaction.depth = 7 và shrinkage = 0.1	0.873	0.237	0.119

Hình 3 hiển thị kết quả của các mô hình hồi quy trên tập huấn luyện (336 mẫu) dựa trên giá trị R^2 và được sắp xếp giảm dần theo khả năng giải thích khác biệt về độ hài lòng giữa các hộ dùng nước. Chúng ta thấy mô hình rừng ngẫu nhiên cho kết quả tốt nhất, giải thích khoảng 93% các khác biệt về độ hài lòng giữa các hộ dùng nước tưới tiêu, theo sát là mô hình boosting có R^2=92.445% và SVR đạt R^2=92.444%. Xếp cuối là phương pháp cây hồi quy có R^2 thấp nhất, khả năng giải thích của mô hình cây hồi quy khoảng 85% kém hơn mô hình hồi quy tuyến tính nhiều biến có R^2=87.481%. Kết quả trên cho thấy mô hình rừng ngẫu nhiên luôn đạt hiệu quả cao nhất dựa vào lỗi dự đoán thấp nhất trên tập dữ liệu kiểm thử và khả năng giải thích mô hình với R^2 tốt nhất.

ả huấn luyện

Kết quả huấn luyện của các mô hình hồi quy dựa trên RMSE so sánh theo từng cặp được trình bày ở Hình 4. Đường kẻ dọc (mốc 0.0) được dùng để làm mốc so sánh, khi hai mô hình hồi quy có lỗi huấn luyện RMSE ngang nhau thì tâm đường thẳng nằm ngang sẽ trùng với mốc. Nếu mô hình ở vị trí bên trái tốt hơn thì tâm đường kẻ ngang lệch sang trái so với mốc, ngược lại sẽ lệch sang phải. Khi hai mô hình hơn kém nhau không đáng kể thì đường kẻ ngang có độ dài ngắn (ví dụ LM-LASSO), ngược lại nếu mô hình hồi quy nổi trội hơn hẳn về lỗi dự đoán thì đường kẻ ngang sẽ kéo dài (chẳng hạn LM-KNN).



Hình 4. So sánh lỗi huấn luyện RMSE của các mô hình hồi quy theo từng cặp.

Hình 5 thể hiện mối quan hệ giữa 34 tiêu chí với giá trị tương quan tuyệt đối từ 0 đến 1. Các tiêu chí có tương quan mạnh với nhau thể hiện bằng kích cỡ hình tròn và màu xanh đậm (đường chéo của biểu đồ là giá trị tương quan của tiêu chí với chính nó). Với những tiêu chí có tương quan yếu hoặc không có quan hệ với nhau sẽ hiển thị trên biểu đồ với màu xanh nhạt và màu trắng tương ứng (tương quan bằng 0). Ta nhận thấy các tiêu chí có ảnh hưởng lẫn nhau đến dịch vụ tưới tiêu thường có tương quan mạnh, chẳng hạn HH1 và DDU8 có thể lý giải là các hệ thống tưới tiêu có chất lượng tốt (HH1) sẽ khắc phục hư hỏng nhanh chóng (DDU8); STC3 và SBD7 có thể hiểu là sự cố công trình được xử lý ngay (STC3) phụ thuộc lớn vào nhân viên thủy lợi khắc phục nhanh (SBD7) v.v. Những số liệu trên biểu đồ tương quan của các tiêu chí rất dễ lý giải trong bài toán thực tế.

Hình 6 hiển thị độ đo sự quan trọng của 34 tiêu chí được sắp xếp theo chiều giảm dần, các độ đo này được tính theo công thức (21) từ rừng ngẫu nhiên. Ta thấy các tiêu chí như HH1, HH7, STC3 có độ quan trọng cao, trong đó HH1=*"Các hệ thống tưới, tiêu có chất lượng tốt, đảm bảo chuyển nước và phân phối nước đến các diện tích cần tưới, tiêu"* có độ quan trọng cao nhất. Kết quả của 3 tiêu chí trên có thể lý giải là trong dịch vụ cung cấp nước tưới tiêu, hộ dùng nước quan tâm nhất đến các hệ thống tưới tiêu có chất lượng tốt, độ đáp ứng của đơn vị cung cấp nước, nó bao gồm những yếu tố như duy tu, bảo dưỡng được thực hiện đầy đủ và đều đặn, sửa chữa sự cố ngay khi công trình hư hỏng hoặc xuống cấp, thực hiện đúng lịch cấp nước, cung cấp tối đa khả năng cấp nước, đáp ứng tốt nhu cầu theo từng giai đoạn sinh trưởng và phát triển của cây trồng, chất lượng nước được đảm bảo.

Cũng trong hình 6, tiêu chí DDU6=*"Nhân viên thủy lợi cung cấp luôn luôn sẵn sàng đáp ứng yêu cầu của ông bà"* có độ quan trọng thấp nhất. Điều này cũng dễ lý giải khi nhân viên thủy lợi có hoặc không đáp ứng những yêu cầu cá nhân của các hộ dùng nước cũng không ảnh hưởng nhiều đến sự hài lòng chung về chất lượng dịch vụ tưới tiêu. Như vậy, mô hình hồi quy ngoài khả năng dự đoán còn trợ giúp người dùng phân tích và hiển thị trực quan các tiêu chí đánh giá, giúp nhà quản lý có thêm thông tin để đầu tư, nâng cấp chất lượng dịch vụ tưới tiêu nhằm đáp ứng cao độ hài lòng của người dân.

IV. KẾT LUẬN

Chúng tôi đã trình bày các mô hình hồi quy dự đoán mức độ hài lòng của các hộ dùng nước liên quan đến dịch vụ tưới tiêu tại đồng bằng Sông Hồng. Các mô hình hồi quy tuyến tính, LASSO, cây hồi quy, k láng giềng, mạng nơron, véctơ hỗ trợ hồi quy, rừng ngẫu nhiên và boosting đã được nghiên cứu, phân tích và so sánh với nhau khi dự đoán độ hài lòng của các hộ dùng nước tưới tiêu dựa trên phương pháp đánh giá R^2, RMSE và MAE. Kết quả thực nghiệm cho thấy mô hình hồi quy tuyến tính tuy dễ cài đặt và dễ sử dụng nhưng lỗi dự đoán cao, các mô hình phi tuyến tỏ ra vượt trội hơn và khả năng dự đoán chính xác hơn, đặc biệt là mô hình rừng ngẫu nhiên cho kết quả dự đoán chính xác nhất và khả năng giải thích khác biệt về biến đích giữa các quan sát tốt nhất. Ngoài ra, độ đo sự quan trọng của các tiêu chí cũng được tính toán từ rừng ngẫu nhiên và hiển thị trực quan giúp nhà quản lý nắm bắt thông tin cần thiết để nâng cấp dịch vụ tưới tiêu. Trong tương lai, chúng tôi sẽ áp dụng kết quả nghiên cứu mở rộng cho các bài toán kinh tế và những bài toán liên quan đến dự đoán với số chiều cao ở Việt Nam.

V. LỜI CẢM ƠN

Xin cảm ơn thầy Đỗ Văn Quang, Phó trưởng Khoa kinh tế và quản lý-Trường Đại học Thủy lợi đã hỗ trợ cung cấp tài liệu và số liệu thử nghiệm.

VI. TÀI LIỆU THAM KHẢO

[1] Christopher M. Bishop et al. Neural networks for pattern recognition. 1995.

[2] J. Martin Bland, Douglas G. Altman, et al. Statistics notes: Cronbach's alpha. *Bmj*, 314(7080):572, 1997.

[3] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[4] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[5] Leo Breiman, Jerome Friedman, Charles J. Stone, and Richard A. Olshen. *Classification and regression trees*. CRC press, 1984.

[6] Yoav Freund, Robert Schapire, and N. Abe. A. short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

[7] Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1):79–103, 1999.

[8] Yoav Freund, Robert E. Schapire, et al. Experiments with a new boosting algorithm. In *ICML*, volume 96, pages 148–156, 1996.

[9] Jerome H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[10] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T. Hastie, J. Friedman, and R. Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

[11] Peter J. Huber. *Robust statistics*. Springer, 2011.

[12] Max Kuhn. Building predictive models in r using the caret package. *Journal of Statistical Software*, 28(5):1–26, 2008.

[13] Thanh Tung Nguyen, Joshua Z. Huang, Qingyao Wu, Thuy T. Nguyen, and Mark J. Li. Genome-wide association data classification and snps selection using two-stage quality-based random forests. *BMC Genomics*, 16(Suppl 2): S5, 2015.

[14] Thanh Tung Nguyen, Joshua Z. Huang, and Thuy Thi Nguyen. Two-level quantile regression forests for bias correction in range prediction. *Machine Learning*, pages 1–19, 2014.

[15] Arun Parasuraman, Leonard L. Berry, and Valarie A. Zeithaml. Refinement and reassessment of the servqual scale. *Journal of retailing*, 1991.

[16] Brian D. Ripley. *Pattern recognition and neural networks*. Cambridge university press, 1996.

[17] Alex J. Smola and Bernhard Schölkopf. A. tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[18] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

PREDICTING THE QUALITY OF IRRIGATION SERVICES IN THE RED RIVER DELTA WITH REGRESSION MODELS

Nguyen Thanh Tung

Faculty of Computer Science and Engineering, Thuyloi University, Hanoi, Vietnam

tungnt@tlu.edu.vn

ABSTRACT *- To predict the satisfaction of users who use the water services is very important for the fee exemption policy to water and agriculture services. This policy has positive impacts on the water exploited and management enterprises, the national budget and social security. In this paper, we present some regression models to predict the satisfaction of users related to the quality of irrigation service in the red river delta. Experimental results showed that the non-linear regression models achieve lower regression errors than linear models. The diversity and feasibility of these regression models can be applied for dealing with economic problems in the domain of water resource management.*

Keywords *- multivariate regression, LASSO, k nearest neighbors, neuron networks, SVR, random forests, data mining, machine learning.*

VII. PHỤ LỤC

Các dòng lệnh chính của ngôn ngữ lập trình R (dùng gói caret) được sử dụng khi tiến hành thực nghiệm.

```
indx <- createFolds(TrainY, k = 5, returnTrain = TRUE)
ctrl <- trainControl(method = "repeatedcv", number = 5, repeats = 2, index = indx)

####### Linear Model##############################
set.seed(1976)
lmTune <- train(x = TrainX, y = TrainY,
                            method = "lm",
                            trControl = ctrl)
lmTune
LM=predict(lmTune, TestX)
testResults<-data.frame(obs=TestY, LM)
print(R2(testResults$obs, testResults$LM))
print(RMSE(testResults$obs, testResults$LM))

set.seed(1976)
lassoTune <- train(x = TrainX, y = TrainY,
                   method = "lasso",
                   trControl = ctrl,
                   preProc = c("center", "scale"))
lassoTune
testResults$Lasso <- predict(lassoTune, TestX)
print(R2(testResults$obs, testResults$Lasso))
print(RMSE(testResults$obs, testResults$Lasso))
plot(lassoTune)

######## Regression Trees#############################################
set.seed(1976)
 cartTune <- train(x = TrainX, y = TrainY,
                    method = "rpart",
                    tuneLength = 25,
                    trControl = ctrl)
cartTune
cartTune$finalModel
testResults$CART <- predict(cartTune$finalModel, TestX)
print(R2(testResults$obs, testResults$CART))
print(RMSE(testResults$obs, testResults$CART))
plot(cartTune, scales = list(x = list(log = 10)))# Plot the tuning results

########### K-Nearest Neighbors#################################
set.seed(1976)
 knnTune <- train(x = TrainX, y = TrainY,
                   method = "knn",
                   preProc = c("center", "scale"),
                   tuneGrid = data.frame(k = 1:20),
                   trControl = ctrl)
knnTune
testResults$KNN <- predict(knnTune, TestX)
print(R2(testResults$obs, testResults$KNN))
print(RMSE(testResults$obs, testResults$KNN))
plot(knnTune)

#################### Support Vector Machines#################################
set.seed(1976)
 svmRTune <- train(x = TrainX, y = TrainY,
                    method = "svmRadial",
                    preProc = c("center", "scale"),
                    tuneLength = 14,
                    trControl = ctrl)
svmRTune
testResults$SVR=predict(svmRTune, TestX)
print(R2(testResults$obs, testResults$SVR))
print(RMSE(testResults$obs, testResults$SVR))
```

```
plot(svmRTune, scales = list(x = list(log = 2))) # Plot the tuning results
############### Random Forests##############################################
mtryGrid <- data.frame(mtry = floor(seq(5, ncol(TrainX)/2, length = 10)))
set.seed(1976)
 rfTune <- train(x = TrainX, y = TrainY,
                 method = "rf",
                 tuneGrid = mtryGrid,
                 ntree = 1000,
                 importance = TRUE,
                 trControl = ctrl)
rfTune
testResults$RF <- predict(rfTune, TestX)
print(R2(testResults$obs, testResults$RF))
print(RMSE(testResults$obs, testResults$RF))
plot(rfTune)# Plot the tuning results

rfImp <- varImp(rfTune, scale = FALSE)
plot(rfImp)

#################### Neural Networks#############################################
nnetGrid <- expand.grid(decay = c(0, 0.01, .1),
                        size = c(1, 3, 5, 7, 9, 11, 13),
                        bag = FALSE)
set.seed(1976)
nnetTune <- train(x = TrainX, y = TrainY,
                  method = "avNNet",
                  tuneGrid = nnetGrid,
                  trControl = ctrl,
                  preProc = c("center", "scale"),
                  linout = TRUE,
                  trace = FALSE,
                  MaxNWts = 13 * (ncol(TrainX) + 1) + 13 + 1,
                  maxit = 1000,
                  allowParallel = FALSE)
nnetTune
testResults$NNet <- predict(nnetTune, TestX)
print(R2(testResults$obs, testResults$NNet))
print(RMSE(testResults$obs, testResults$NNet))
plot(nnetTune)

############### Boosting####################################################
gbmGrid <- expand.grid(interaction.depth = seq(1, 7, by = 2),
                       n.trees = seq(100, 1000, by = 50),
                       shrinkage = c(0.01, 0.1))
set.seed(100)
gbmTune <- train(x = TrainX, y = TrainY,
                 method = "gbm",
                 tuneGrid = gbmGrid,
                 trControl = ctrl,
                 verbose = FALSE)
gbmTune
testResults$GBM <- predict(gbmTune, TestX)
print(R2(testResults$obs, testResults$GBM))
print(RMSE(testResults$obs, testResults$GBM))
plot(gbmTune, auto.key = list(columns = 4, lines = TRUE))

gbmImp <- varImp(gbmTune, scale = FALSE)
plot(gbmImp)
```