# PHÂN LỚP DỮ LIỆU MẤT CÂN BẰNG VỚI THUẬT TOÁN HBU

*NGUYỄN THỊ LAN ANH*
*Khoa Tin học, Trường Đại học Sư phạm, Đại học Huế*

**Tóm tắt:** Dữ liệu mất cân bằng là một trong những nguyên nhân làm giảm hiệu suất của bài toán phân lớp. Nhiều phương pháp đã được nghiên cứu để giải quyết vấn đề này. Trong bài báo này chúng tôi đề xuất một thuật toán làm giảm số lượng phần tử lớp đa số, đặc biệt là các phần tử ở đường biên, dựa trên Hypothesis margin của các đối tượng thuộc lớp thiểu số để cải thiện hiệu suất phân lớp tập dữ liệu mất cân bằng.

**Từ khóa**: Dữ liệu mất cân bằng, phương pháp làm giảm số lượng phần tử, Hypothesis margin

## 1. GIỚI THIỆU

Khi một tập dữ liệu có số lượng phần tử thuộc một hoặc một số nhãn lớp lớn hơn số lượng phần tử thuộc các nhãn lớp còn lại, tập dữ liệu đó được gọi là mất cân bằng. Đối với bài toán phân lớp hai lớp tập dữ liệu bị mất cân bằng, lớp có số lượng phần tử nhiều hơn gọi là lớp đa số, lớp có số phần tử ít hơn gọi là lớp thiểu số. Đây cũng là loại bài toán chúng tôi đề cập đến trong bài báo này.

Nghiên cứu về dữ liệu mất cân bằng, trong những năm gần đây, là một trong những vấn đề quan tâm của nhiều nhà khoa học trong nước cũng như trên thế giới bởi tính thực tế và phổ biến của nó. Bài toán phân lớp dữ liệu mất cân bằng nhằm mục đích phát hiện các đối tượng hiếm nhưng quan trọng, và được ứng dụng trong nhiều lĩnh vực khác nhau như phát hiện gian lận tài chính, dự đoán cấu trúc protein, dự đoán tương tác giữa protein-protein, phân lớp microRNA…, hay chẩn đoán bệnh trong y học. Dữ liệu mất cân bằng làm giảm hiệu quả của các thuật toán phân lớp truyền thống vì các bộ phân lớp này có khuynh hướng dự đoán lớp đa số và bỏ qua lớp thiểu số [1]. Hay nói cách khác, hầu hết các phần tử thuộc lớp đa số sẽ được phân lớp đúng và các phần tử thuộc lớp thiểu số cũng sẽ được gán nhãn lớp là nhãn lớp của lớp đa số, kết quả là độ chính xác (Accuracy) của việc phân lớp rất cao trong khi độ nhạy (Sensitivity) lại rất thấp.

Nhiều phương pháp nâng cao hiệu quả bài toán phân lớp dữ liệu mất cân bằng đã được đề xuất, bao gồm các phương pháp tiếp cận ở mức độ thuật toán như điều chỉnh xác suất ước lượng, sử dụng các hằng số phạt khác nhau cho các nhãn lớp khác nhau [2]; các phương pháp tiếp cận ở mức dữ liệu như sinh thêm các phần tử cho lớp thiểu số [3],[4], giảm bớt các phần tử thuộc lớp đa số [5]; và nhóm các phương pháp kết hợp. Một số tác giả đã chỉ ra rằng các phương pháp tiếp cận ở mức dữ liệu hiệu quả hơn các phương pháp còn lại trong việc cải thiện độ chính xác sự phân lớp các tập dữ liệu mất cân bằng [2].

Phương pháp sinh phần tử đơn giản nhất là sinh phần tử ngẫu nhiên (Random Oversampling). Phương pháp này làm tăng số lượng phần tử lớp thiểu số bằng cách

Tạp chí Khoa học, Trường Đại học Sư phạm, Đại học Huế
ISSN 1859-1612, Số 04(48)/2018: tr. 110-116
Ngày nhận bài: 23/11/2017; Hoàn thành phản biện: 22/12/2017; Ngày nhận đăng: 11/01/2018

chọn ngẫu nhiên một số phần tử thuộc lớp này và nhân bản chúng để làm giảm tỷ lệ mất cân bằng. Nhược điểm của kỹ thuật này là dễ dẫn đến tình trạng quá khớp với dữ liệu huấn luyện (overfitting). Hơn nữa, nếu tập dữ liệu có kích thước lớn thì chi phí thời gian và bộ nhớ cho giai đoạn phân lớp sẽ gia tăng đáng kể.

Phương pháp giảm số phần tử ngẫu nhiên (Random Undersampling) là phương pháp làm giảm phần tử lớp đa số đơn giản nhất bằng cách ngẫu nhiên chọn và loại bỏ một số phần tử thuộc lớp đa số. Phương pháp này tuy tốn ít chi phí về thời gian cũng như bộ nhớ cho quá trình phân lớp nhưng lại dễ làm mất các thông tin quan trọng của lớp đa số. Để khắc phục nhược điểm này, thay vì chọn ngẫu nhiên, HMU [5] chọn các phần tử có giá trị lề giả thuyết bé nhất để loại bỏ. Phương pháp này đã làm tăng đáng kể hiệu quả của việc phân lớp. Tuy nhiên, trong một số trường hợp, khi các phần tử thuộc hai lớp đa số và thiểu số nằm gần nhau, đặc biệt là khi các phần tử lớp đa số phân tán xen giữa các phần tử lớp thiểu số, các phần tử này sẽ dễ bị phân lớp nhầm và thuật toán HMU sẽ không hoạt động tốt.

Trong bài báo này, một phương pháp làm giảm số phần tử thuộc lớp đa số mới được đề xuất nhằm xử lý các đối tượng khó phân lớp và khắc phục nhược điểm đã đề cập.

## 2. LỀ GIẢ THUYẾT

Lề (Margin), đóng vai trò quan trọng trong lĩnh vực học máy, thể hiện khả năng phân lớp của bộ phân lớp (classifier). Có thể xác định lề cho một phần tử dựa trên quy tắc phân lớp bằng cách đo khoảng cách từ phần tử đang xét tới biên quyết định được xác định bởi bộ phân lớp; hoặc tính khoảng cách mà bộ phân lớp có thể di chuyển sao cho không làm thay đổi nhãn lớp của các phần tử đã được xác định [6]. Lề được đề cập đến ở cách thứ nhất gọi là lề đối tượng (Sample margin) và theo cách còn lại gọi là lề giả thuyết (Hypothesis margin).

Khi sử dụng bộ phân lớp láng giềng gần nhất, các kết quả sau đây được thừa nhận là đúng [7]:

1. Lề giả thuyết là giới hạn dưới của lề đối tượng.

2. Lề giả thuyết của phần tử x trong tập dữ liệu A được tính bởi công thức:

$$\theta_A = \frac{1}{2}(\|x - \text{nearestmiss}_A(x)\| - \|x - \text{nearesthit}_A(x)\|) \quad (1)$$

trong đó: $\text{nearesthit}_A(x)$ là phần tử gần nhất có cùng nhãn lớp với x trong A.

$\text{nearestmiss}_A(x)$ là phần tử gần nhất khác nhãn lớp với x trong A.

Từ đó có thể suy ra, nếu một tập các phần tử có lề giả thuyết lớn thì giá trị lề đối tượng tương ứng của nó cũng lớn.

Ngoài ra, chúng tôi đã chứng minh được rằng nếu loại bỏ một phần tử thuộc lớp đa số sẽ làm tăng giá trị lề của các phần tử lớp thiểu số và giảm giá trị lề của phần tử thuộc lớp đa số, nghĩa là việc chọn các phần tử có giá trị lề giả thuyết bé nhất thay vì chọn một cách

ngẫu nhiên để loại bỏ sẽ làm tăng hiệu suất của việc phân lớp. Do đó, thuật toán HMU đã được đề xuất để làm tăng hiệu suất của việc phân lớp dữ liệu mất cân bằng [5].

Tuy nhiên, trong một số trường hợp, khi các phần tử lớp đa số nằm phân tán giữa các phần tử lớp thiểu số, tức là nằm ở gần biên quyết định, sử dụng HMU khó có thể chọn các phần tử này để loại bỏ được. Trong khi đó, các phần tử nằm càng xa biên lại càng dễ được phân lớp đúng [8].

## 3. PHƯƠNG PHÁP LÀM GIẢM PHẦN TỬ HBU

Dựa vào những phân tích ở phần trên, chúng tôi đề xuất một phương pháp mới để xử lý bài toán phân lớp dữ liệu mất cân bằng là phương pháp làm giảm phần tử dựa vào giá trị lề giả thuyết và ưu tiên loại bỏ các phần tử nằm ở biên, đặt tên là HBU (Hypothesis margin based Borderline Under-sampling). Với mỗi phần tử lớp thiểu số x, một số lượng n các phần tử lớp đa số nằm gần x nhất sẽ được chọn để loại bỏ. Giá trị n thay đổi phụ thuộc vào giá trị lề của mỗi x khác nhau.

Thuật toán được mô tả như sau:

***HBU Algorthim***

*Input: tập các phần tử lớp thiểu số P; tập các phần tử lớp đa số N; số lượng phần tử cần loại bỏ N' ; tham số k*

*Output: tập các phần tử lớp đa số mới N*;*

1. *Begin*
2. *Tính giá trị lề mar(x) của tất cả các phần tử lớp thiểu số x trên tập dữ liệu đã cho*
3. *max =* $\max_{x \in P} mar(x)$
4. *min =* $\min_{x \in P} mar(x)$
5. *p = |P|*
6. *Foreach x in P*
7. ***nos****= int((N'/p)*(k +(max-mar(x))/(max-min)));*
8. *Loại bỏ* ***nos*** *phần tử lớp đa số mà gần với x nhất*
9. *N' = N' - nos*
10. *p = p - 1*
11. *End-for*
12. *End*

Lề của các phần tử lớp đa số được tính dựa vào công thức (1). Trong phạm vi bài báo này, chúng tôi sử dụng khoảng cách Euclide để xác định giá trị lề cho các đối tượng.

Kích thước của lớp đa số sau khi làm giảm bớt số phần tử N* được xác định dựa vào số lượng phần tử cần loại bỏ N', giá trị này phụ thuộc vào từng tập dữ liệu cụ thể.

## 4. THỰC NGHIỆM ĐÁNH GIÁ HIỆU SUẤT THUẬT TOÁN

Chúng tôi tiến hành thực nghiệm trên 4 tập dữ liệu UCI [9] là Balance, Cmc, Haberman và Pima để đánh giá hiệu suất của quá trình phân lớp. Bảng 1 mô tả thông tin về số

lượng thuộc tính, số phần tử, tỷ lệ mất cân bằng (số phần tử tập thiểu số:số phần tử tập đa số) của các tập dữ liệu này. Hàm normalize của gói lệnh SOM trong R được dùng để normalize tất cả các tập dữ liệu trước khi tiến hành điều chỉnh tỷ lệ mất cân bằng cũng như thực hiện phân lớp.

Bảng 1. *Các tập dữ liệu UCI*

| Tập dữ liệu | Số thuộc tính | Số phần tử | Tỷ lệ mất cân bằng |
|---|---|---|---|
| Balance | 4 | 625 | 1:11.75 |
| Cmc | 9 | 1473 | 1:3.42 |
| Haberman | 3 | 306 | 1:2.78 |
| Pima | 8 | 768 | 1:1.87 |

Trong bài báo này, chúng tôi sử dụng gói lệnh kernlab [10] trong R cho việc phân lớp để so sánh kết quả phân lớp bộ dữ liệu gốc không có can thiệp của thuật toán làm thay đổi số phần tử để xử lý sự mất cân bằng dữ liệu (ORIginal), kết quả phân lớp có sử dụng thuật toán giảm số phần tử ngẫu nhiên (RUS), kết quả có sử dụng thuật toán HMU với kết quả khi sử dụng thuật toán HBU nhằm đánh giá tính hiệu quả của thuật toán này.

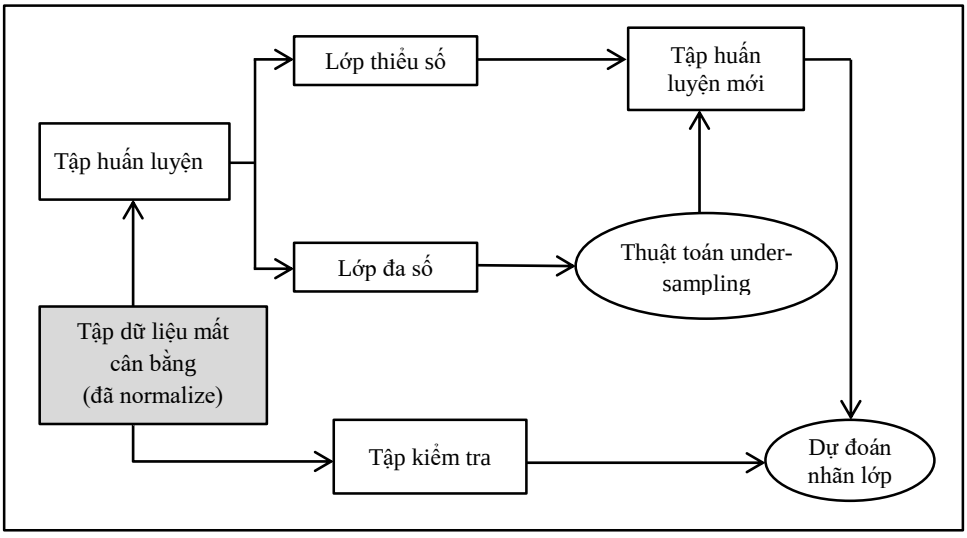
Quá trình phân lớp được thực hiện như sau:

- Máy vector hỗ trợ (Support Vector Machine - SVM) được sử dụng làm bộ phân lớp chính.

- Chúng tôi thực hiện mười lần kiểm chứng chéo 10-fold (10-fold cross-validation) cho mỗi bộ dữ liệu. Sau đó, các giá trị độ đo đánh giá hiệu suất được tính bằng cách lấy giá trị trung bình cộng của mười lần thực hiện độc lập này.

- Sau khi áp dụng các thuật toán điều chỉnh tỷ lệ mất cân bằng, các tập dữ liệu mới có tỷ lệ số phần tử lớp thiểu số:lớp đa số là xấp xỉ 1:1.

Hình 1 bên dưới mô tả quá trình phân lớp đánh giá hiệu suất thuật toán trên các tập dữ liệu.




Hình 1. *Quá trình thực nghiệm phân lớp dữ liệu*

Trong trường hợp tập dữ liệu là cân bằng, độ chính xác thường được sử dụng để đánh giá hiệu quả của quá trình phân lớp. Tuy nhiên, khi phân lớp tập dữ liệu không cân bằng, độ chính xác toàn thể này không còn thích hợp để xác định tính hiệu quả của mô hình phần lớp nữa. Khi tỷ lệ mất cân bằng trong tập dữ liệu lớn, các bộ phân lớp thông thường sẽ cho ra độ chính xác rất cao vì hầu như tất cả các phần tử đều được gán nhãn lớp là lớp đa số và rất hiếm phần tử được gán nhãn lớp của lớp thiểu số. Do đó, một số độ đo khác như độ nhạy, tính đặc trưng (Specificity), G-trung bình (G-mean), F-measure, MCC… được sử dụng làm độ đo hiệu suất phân lớp dữ liệu mất cân bằng. Trong phạm vi bài báo này, chúng tôi sử dụng các độ đo sau:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \tag{2}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \tag{3}$$

$$\text{G} - \text{mean (Balanced accuracy)} = \sqrt{\text{Sensitivity} \times \text{Specificity}} \tag{4}$$

Ở đây, TP và FN lần lượt là số phần tử lớp thiểu số được dự đoán đúng và bị dự đoán sai so với nhãn lớp thực sự của chúng; TN và FP lần lượt là số phần tử lớp đa số được dự đoán đúng và sai so với nhãn lớp thực sự của chúng. Các giá trị TP, TN, FP, FN được xác định dựa vào ma trận nhầm lẫn (confusion matrix) như ở bảng 2 bên dưới.

Bảng 2. *Ma trận nhầm lẫn*

| | **Thực tế thuộc lớp thiểu số** | **Thực tế thuộc lớp đa số** |
|---|---|---|
| **Dự đoán thuộc lớp thiểu số** | **TP** (Thuộc lớp thiểu số được dự đoán là thuộc lớp thiểu số) | **FP** (Thuộc lớp đa số được dự đoán là thuộc lớp thiểu số) |
| **Dự đoán thuộc lớp đa số** | **FN** (Thuộc lớp thiểu số được dự đoán là thuộc lớp đa số) | **TN** (Thuộc lớp đa số được dự đoán là thuộc lớp đa số) |

## 5. KẾT QUẢ THỰC NGHIỆM

Bảng 3 và Bảng 4 trình bày kết quả đánh giá hiệu suất của các phương pháp khác nhau trên bốn tập dữ liệu UCI là Balance, Cmc, Haberman và Pima theo các độ đo Sensitivity, Specificity và G-mean.

So sánh kết quả khi áp dụng các phương pháp làm giảm phần tử với kết quả của phương pháp ORI cho thấy hiệu quả của các phương pháp này trong xử lý sự mất cân bằng dữ liệu. Trong khi tỷ lệ số phần tử lớp thiểu số được dự đoán đúng bằng ORI thấp, tỷ lệ này tăng lên đáng kể, trên 50%, khi áp dụng các phương pháp làm giảm phần tử. Ví dụ như đối với tập dữ liệu Balance có tỷ lệ mất cân bằng giữa lớp thiểu số và đa số khá lớn (1:11.75), ORI hoàn toàn không phân lớp đúng được một phần tử lớp thiểu số nào khi Sensitivity = 0%, thì giá trị này đã tăng lên 77.76% chỉ bằng cách sử dụng phương pháp làm giảm phần tử ngẫu nhiên RUS. Hay đối với các tập dữ liệu Cmc, Pima và Haberman, Sensitivity của ORI tăng lần lượt là 60.24%, 19.23%, 28.76% khi áp dụng

RUS. So với ORI, G-mean của phương pháp HMU trên cả bốn tập dữ liệu Balance, Cmc, Haberman, Pima đều tăng lần lượt là 58.47%, 41.27%, 16.3%, 3.4%.

Kết quả thể hiện ở bảng 3 và bảng 4 cũng cho thấy cải tiến của phương pháp làm giảm phần tử mới HBU hoạt động có hiệu quả so với phương pháp HMU và RUS. Đối với các tập dữ liệu Balance, Cmc và Pima, giá trị G-mean của HBU cũng đã tăng so với G-mean của HMU (0.86%, 0.28% và 0.96%).

Bảng 3. *Kết quả phân lớp theo độ đo Sensitivity (%) và Specificity (%)của các tập dữ liệu UCI*

| Tập dữ liệu | | Balance | Haberman | Cmc | Pima |
|---|---|---|---|---|---|
| **Sensitivity (%)** | ORIginal | 0.00 | 23.46 | 6.04 | 56.16 |
| | RUS | 77.76 | 52.22 | 66.28 | 75.34 |
| | HMU | **99.80** | 63.33 | **75.35** | **78.06** |
| | HBU | 87.56 | **67.53** | 67.03 | 70.82 |
| **Specificity (%)** | ORIginal | **100.00** | **93.02** | **98.62** | **87.74** |
| | RUS | 43.98 | 76.04 | 65.07 | 71.90 |
| | HMU | 35.71 | 71.73 | 57.81 | 69.06 |
| | HBU | 41.89 | 62.89 | 65.54 | 78.12 |

Bảng 4. *Kết quả phân lớp theo độ đo G-mean (%) của các tập dữ liệu UCI*

| Tập dữ liệu | Balance | Haberman | Cmc | Pima |
|---|---|---|---|---|
| ORIginal | 0.00 | 46.71 | 24.40 | 70.19 |
| RUS | 58.47 | 63.01 | 65.67 | 73.59 |
| HMU | 59.70 | **67.39** | 66.00 | 73.42 |
| HBU | **60.56** | 65.17 | **66.28** | **74.38** |

## 6. KẾT LUẬN

Trong bài báo này, chúng tôi đã trình bày một thuật toán làm giảm phần tử lớp đa số mới HBU và thực hiện đánh giá hiệu suất của thuật toán này và so sánh với các thuật toán khác trên bốn tập dữ liệu chuẩn UCI. Kết quả thực nghiệm đã cho thấy thuật toán được đề xuất có hiệu quả trên bốn tập dữ liệu UCI dựa trên các giá trị độ đo đánh giá hiệu suất Sensitivity, Specificity và G-mean. Phương pháp này có thể kết hợp với các phương pháp làm tăng phần tử khác hoặc các phương pháp lựa chọn đặc trưng để cho kết quả tốt hơn, đặc biệt là đối với các tập dữ liệu có kích thước lớn.

## TÀI LIỆU THAM KHẢO

[1] Sain, H. & Purnami, S. W. (2015). Combine Sampling Support Vector Machine for Imbalanced Data Classification. *Procedia Comput. Sci.* **72,** 59–66.

[2] He, H. & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Trans. Knowl. Data Eng.* **21,** 1263–1284.

[3] Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. (2002). SMOTE : Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **16,** 321–357.
[4] Nguyen, L. A. T. *et al* (2013). Improving the Prediction of Protein-Protein Interaction Sites Using a Novel Over-Sampling Approach and Predicted Shape Strings. *Annu. Rev. Res. Biol.* **3,** 92–106.
[5] Nguyễn Thị Lan Anh (2017). Thuật toán HMU trong bài toán phân lớp dữ liệu mất cân bằng. *Tạp chí Khoa học và Giáo dục, Trường Đại học Sư phạm Huế* **2,** 101–108.
[6] Crammer, K., Gilad-Bachrach, R., Navot, A. & Tishby, N. (2002). Margin Analysis of The Lvq Algorithm. *Neural Inf. Process. Syst.* 462–469.
[7] Ran, G., Amir, N. & Naftali, T. (2004). Margin Based Feature Selection - Theory and Algorithms. in *Proceedings of the 21st International Conference on Machine Learning*.
[8] Anand, A., Pugalenthi, G., Fogel, G. B. & Suganthan, P. N. (2010). An approach for classification of highly imbalanced data using weighting and undersampling. *Amino Acids* **39,** 1385–1391.
[9] Lichman, M. (2013). UCI Machine Learning Repository. *http://archive.ics.uci.edu/ml*, Irvine, CA: University of California, School of Information and Computer Science.
[10] Karatzoglou, A., Wien, T. U., Smola, A., Hornik, K. & Wien, W. (2004). kernlab – An S4 Package for Kernel Methods in R. *J. Stat. Softw.* **11,** 1–20.

**Title:** HYPOTHESIS MARGIN BASED BORDERLINE UNDERSAMPLING METHOD FOR CLASSIFYING IMBALANCED DATA SETS

**Abstract:** Classifying imbalanced data is one of the challenges in machine learning. Therefore, many techniques have been developed to handle this problem. In this paper, we propose an under-sampling method that is based on the hypothesis margin of minority class samples to remove the majority samples nearby the borderline. The experimental results show that our approach is comparable to some other methods in terms of G-mean.

**Keywords**: Imbalanced data, Under-sampling, Hypothesis margin.