NÂNG CAO HIỆU QUẢ PHÂN LỚP DỮ LIỆU KHÔNG CÂN BẰNG SỬ DỤNG KỸ THUẬT TĂNG MẪU THIỂU SỐ VÀ ĐẶC TRƯNG CỦA MỖI CỤM

Phan Anh Phong*, Lê Văn Thành
Trường Đại học Vinh, Nghệ An, Việt Nam

ARTICLE INFORMATION

Journal: *Vinh University Journal of Science Natural Science, Engineering and Technology*
p-ISSN*: 3030-4563*
e-ISSN: *3030-4180*

Volume: 53
Issue: 3A
****Correspondence:***
phongpa@gmail.com

Received: *19 April 2024*
Accepted: *21 June 2024*
Published: *20 September 2024*

Citation:
Phan Anh Phong, Le Van Thanh (2024). Improving performance for imbalanced data classification using oversampling and characteristics of each cluster
Vinh Uni. J. Sci.
Vol. 53 (3A), pp. 5-15
doi: 10.56824/vujs.2024a054a

TÓM TẮT

Bài báo đề xuất một phương pháp để nâng cao hiệu quả phân lớp dữ liệu không cân bằng. Đóng góp chính của phương pháp là kết hợp thuật toán phân cụm K-means và kỹ thuật sinh mẫu thiểu số VCIR để tạo ra các mẫu nhân tạo có tính đại diện sát với đặc trưng của dữ liệu thực tế. Các kết quả thực nghiệm đã chỉ ra rằng phương pháp đề xuất đạt hiệu quả cao hơn trên một số độ đo so với các phương pháp xử lý dữ liệu không cân bằng phổ biến hiện nay như SMOTE, Borderline-SMOTE, Kmeans-SMOTE và SVM-SMOTE.

Từ khóa: Phân lớp dữ liệu; dữ liệu không cân bằng; oversampling; K-Means; SMOTE.

1. Giới thiệu

Phân lớp dữ liệu là một bài toán quan trọng trong học máy, đã và đang được ứng dụng ở nhiều lĩnh vực của đời sống xã hội [2]-[3]. Trong thực tế, nhiều trường hợp dữ liệu thu thập để xây dựng các mô hình phân lớp thường không cân bằng nhãn lớp. Đó là hiện tượng khi số lượng mẫu dữ liệu của một hoặc một số lớp (gọi là lớp thiểu số) ít hơn nhiều so với số lượng mẫu dữ liệu của các lớp khác (gọi là lớp đa số) [1]. Bài toán phân lớp trên tập dữ liệu không cân bằng, đặc biệt là phân lớp nhị phân (có hai nhãn lớp) xuất hiện khá phổ biến, ví dụ như: Phát hiện gian lận thẻ tín dụng (số lượng giao dịch gian lận thường ít hơn nhiều so với số lượng giao dịch hợp lệ) [2]; Chẩn đoán bệnh (số lượng người bị bệnh thường ít hơn so với số lượng người đến khám); Phân loại email rác (số lượng email rác thường ít hơn nhiều so với số lượng email bình thường) [3].

Khi tỉ lệ không cân bằng của bộ dữ liệu cao thì các mô hình phân lớp thường nhận diện kém các phần tử ở lớp thiểu số, đây là những phần tử quan trọng trong các ứng dụng. Hay nói một cách khác, mô hình phân lớp truyền thống sẽ hoạt động kém hiệu quả trên các bộ dữ liệu không cân bằng [4], [14]. Hiện nay có hai hướng tiếp cận chính để nâng cao hiệu quả của bài toán phân lớp dữ liệu không cân bằng, bao gồm hướng tiếp cận theo dữ liệu và theo giải thuật [14]. Ở hướng tiếp cận thứ nhất,

các giải pháp tập trung vào việc điều chỉnh, cải tiến các giải thuật phân lớp truyền thống như Decision Tree, KNN, SVM… sao cho mô hình có hiệu quả cao đối với các mẫu trong lớp thiểu số như phương pháp điều chỉnh xác suất ước lượng đối với Decision Tree [5], bổ sung hằng số thưởng hoặc phạt cho mỗi lớp hoặc điều chỉnh ranh giới phân lớp đối với SVM [6]. Hướng tiếp cận thứ hai, các phương pháp hướng tới điều chỉnh sự không cân bằng của dữ liệu bằng cách áp dụng kỹ thuật sinh thêm phần tử ở lớp thiểu số (Over-sampling) hoặc giảm phần tử ở lớp đa số (Under-sampling), với các kỹ thuật phổ biến như SMOTE [7], ADASYN [8], Tomek links [9]. Ngoài ra, cũng có thể kết hợp cả hai phương pháp trên để cùng lúc giảm phần tử ở lớp đa số và tăng phần tử ở lớp thiểu số.

Đối với phương pháp sinh mẫu ở lớp thiểu số, SMOTE và các biến thể của nó như BorderlineSMOTE [10], SVM-SMOTE [11]... là các kỹ thuật có hiệu quả cao và được sử dụng khá rộng rãi. Kỹ thuật sinh mẫu trong SMOTE được mô tả ngắn gọn như sau: với mỗi mẫu x của lớp thiểu số, chọn ngẫu nhiên một trong số k láng giềng gần nhất cùng nhãn lớp với x và sinh mẫu nhân tạo trên đoạn thẳng nối mẫu đang xét và láng giềng được lựa chọn [7]. Trong BorderlineSMOTE, các mẫu lớp thiểu số được chia thành 3 nhóm: nhiễu, đường biên và an toàn, bằng cách tính toán số mẫu thuộc lớp đa số trong k lân cận gần nhất, sau đó tiến hành sinh mẫu mới tương tự SMOTE nhưng chỉ thực hiện đối với các mẫu nằm trên đường biên [10]. SVM-SMOTE tập trung vào việc tăng các mẫu thiểu số gần đường biên bằng mô hình SVM để giúp thiết lập đường biên giữa các lớp, với lập luận rằng các trường hợp xung quanh đường biên là rất quan trọng [11]. Đối với Kmeans-SMOTE, các mẫu được phân cụm theo thuật toán K-Means, sau đó chọn các cụm có tỉ lệ chênh lệch cao (lớn hơn 50%) và tiến hành sinh mẫu mới trên các cụm đó tương tự SMOTE, số lượng mẫu mới được sinh ra dựa trên độ thưa thớt của lớp thiểu số trong cụm, nếu cụm càng thưa thớt, các mẫu sinh ra càng nhiều [12].

Các kỹ thuật sinh mẫu thiểu số trên đây đều dựa vào SMOTE, tuy nhiên, SMOTE thường có nhược điểm, mẫu mới được tạo ra không có tính đại diện cao cho dữ liệu thực tế và thường nhạy cảm với nhiễu. Hiện nay có một số kỹ thuật sinh mẫu thiểu số không dùng SMOTE, chẳng hạn kỹ thuật CIR trong [13]. Quy trình sinh mẫu của CIR được mô tả như sau: Trước tiên, chọn tâm C từ các mẫu thiểu số, đó là điểm trung bình của các mẫu này; tiếp theo, tìm mẫu thiểu số gần tâm C nhất, ký hiệu D_{min} và cuối cùng là sinh ra các mẫu nhân tạo $D_j = D_{min} + h_j \times C$, với h_j là một giá trị thuộc (0, 1).

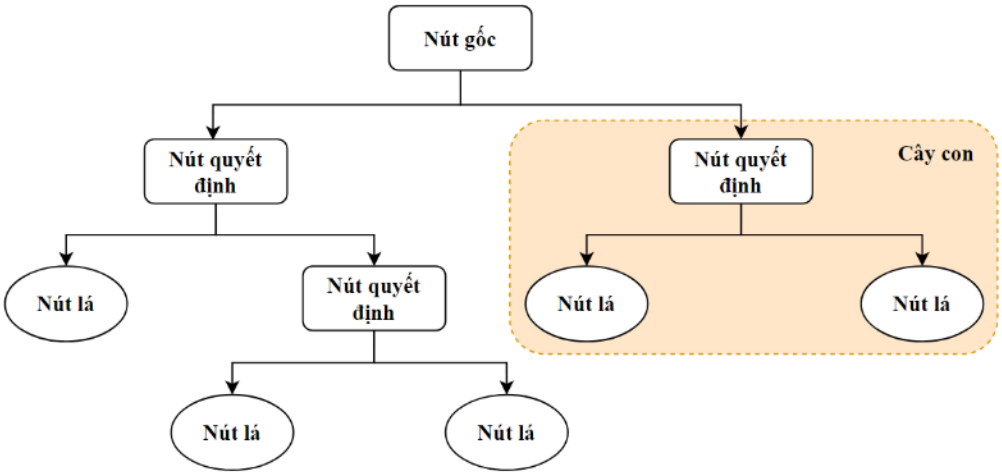
Bài báo này đề xuất một phương pháp để nâng cao hiệu quả phân lớp dữ liệu không cân bằng. Điểm mới của phương pháp là sự kết hợp thuật toán phân cụm K-means và kỹ thuật sinh mẫu thiểu số để tạo ra các mẫu nhân tạo có tính đại diện sát với đặc trưng của dữ liệu thực tế. Phần còn lại của bài báo được bố cục như sau: Phần 2 giới thiệu vắn tắt một số thuật toán phân lớp tiêu biểu; Phần 3 trình bày phương pháp đề xuất; Phần 4 là kết quả thực nghiệm của mô hình trên các độ đo thường được sử dụng để đánh giá các mô hình phân lớp với tập dữ liệu không cân bằng trong y tế; Cuối cùng là kết luận bài báo và một số hướng phát triển tiếp theo.

2. Một số thuật toán phân lớp tiêu biểu

Phần này giới thiệu sơ qua về ba thuật toán phân lớp phổ biến là Decision Tree, KNN (K-Nearest Neighbors) và SVM (Support Vector Machine). Các thuật toán này được sử dụng trong các thử nghiệm ở phần 4 của bài báo.

2.1. Decision Tree

Thuật toán Cây quyết định (Decision Tree - DT) là một thuật toán học có giám sát được sử dụng cho cả bài toán phân lớp và hồi quy. DT sử dụng một cấu trúc dạng cây để mô hình hóa mối quan hệ giữa các thuộc tính (đặc trưng) và nhãn lớp của dữ liệu. Về cấu trúc, một cây quyết định bao gồm các nút (node) và cạnh (Hình 1). Nút là đại diện cho một quyết định và cạnh là đại diện cho một điều kiện để phân chia dữ liệu. Mỗi cạnh có một ngưỡng giá trị để chia dữ liệu thành các nhánh con. Có hai loại nút chính: Nút gốc (root node) là nút đầu tiên của cây, đại diện cho toàn bộ tập dữ liệu; Nút lá (leaf node): là nút cuối cùng của cây, đại diện cho một nhãn lớp cụ thể. Các biến thể phổ biến của thuật toán cây quyết định bao gồm ID3, C4.5 và CART.



Hình 1: *Minh họa thuật toán Decision Tree*

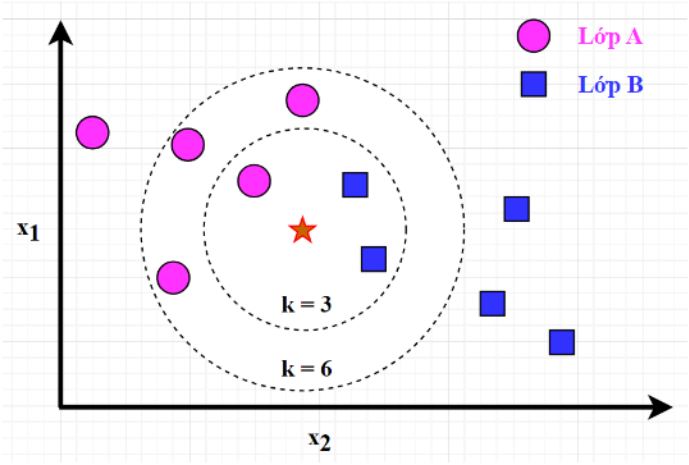
2.2. KNN (K-Nearest Neighbors)

Thuật toán K láng giềng gần nhất (K-Nearest Neighbors - KNN) là thuật toán học máy có giám sát. Ý tưởng chính của KNN là dựa vào sự tương đồng của các điểm dữ liệu. Khi các điểm dữ liệu có xu hướng thuộc về cùng một lớp nếu chúng tương tự nhau, hay nói cách khác là chúng có khoảng cách gần nhau trong không gian đặc trưng.

Giả sử ta có tập dữ liệu huấn luyện được chia thành các lớp và có một điểm dữ liệu mới cần phân lớp điểm đó thuộc lớp nào. Khi đó các bước cơ bản của thuật toán KNN được mô tả như sau:

- Bước 1: Tính khoảng cách giữa điểm dữ liệu mới này với tất cả các điểm dữ liệu trong tập dữ liệu huấn luyện. Khoảng cách thường được tính bằng các độ đo phổ biến như khoảng cách Euclid hoặc Manhattan.
- Bước 2: Chọn ra k điểm dữ liệu gần nhất với điểm dữ liệu mới, trong đó k là một số nguyên dương cho trước.
- Bước 3: Dựa trên nhãn lớp của k láng giềng gần nhất, KNN sẽ gán nhãn lớp cho điểm dữ liệu mới theo nhãn lớp phổ biến nhất trong số k láng giềng đó.

Hình 2 minh họa thuật toán KNN theo các giá trị k khác nhau. Khi $k = 3$ thì điểm dữ liệu mới (hình sao) thuộc lớp nhãn hình vuông, khi $k = 6$ thì lại thuộc lớp nhãn hình tròn.

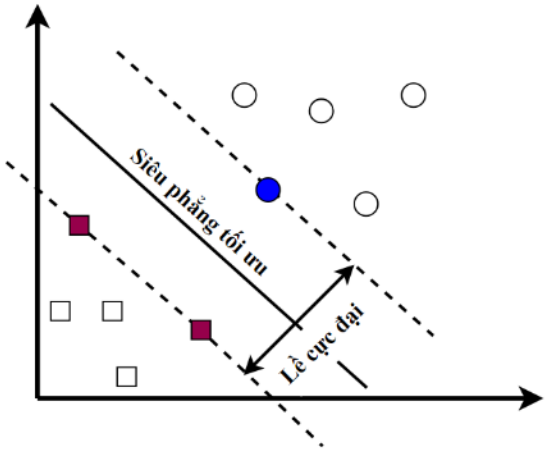


Hình 2: *Minh họa thuật toán KNN với giá trị k khác nhau*

2.3. SVM (Support Vector Machine)

Thuật toán SVM (Support Vector Machine) là một thuật toán học máy có giám sát được sử dụng phổ biến cho các bài toán phân lớp [14]-[15]. Mục tiêu của SVM là tìm siêu phẳng phân chia tối ưu dữ liệu trong không gian đặc trưng để phân tách các điểm dữ liệu thuộc các lớp khác nhau. Nói một cách khác, SVM cố gắng tìm một ranh giới có thể tách biệt các nhóm dữ liệu một cách tốt nhất, giảm thiểu sai sót trong việc phân lớp. Hoạt động của SVM được mô tả như sau:

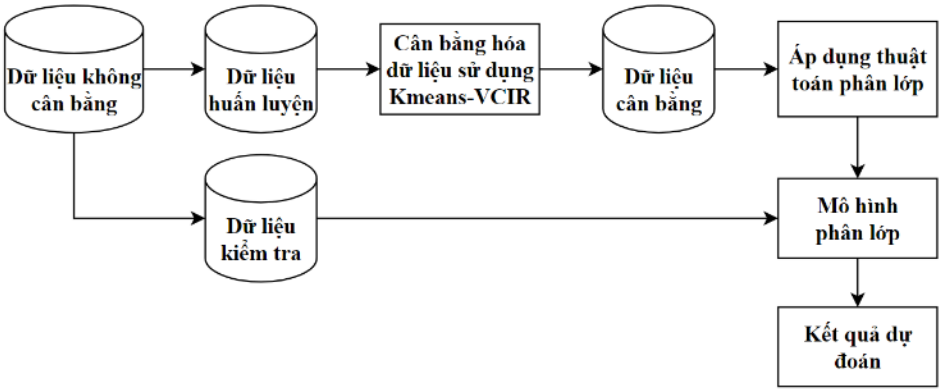
- Mỗi điểm dữ liệu được biểu diễn như một vectơ trong không gian đa chiều, mỗi chiều tương ứng với một thuộc tính của tập dữ liệu.
- Tìm một siêu phẳng sao cho nó có thể phân chia các điểm dữ liệu thuộc các lớp khác nhau một cách tối ưu nhất. "Tối ưu" ở đây có nghĩa là khoảng cách giữa siêu phẳng tới các điểm dữ liệu ở các lớp gần nhất là lớn nhất.
- Khi có một điểm dữ liệu mới, SVM sẽ dự đoán lớp của nó dựa vào vị trí của điểm này so với siêu phẳng đã được tìm ra.



Hình 3: *Minh họa siêu phẳng tối ưu*

3. Phương pháp đề xuất

Phần này đề xuất phương pháp để nâng cao hiệu quả phân lớp dữ liệu không cân bằng. Đóng góp chính của phương pháp là kết hợp thuật toán phân cụm K-means và kỹ thuật sinh mẫu thiểu số VCIR (Class Imbalance Reduction) để tạo ra các mẫu nhân tạo. Hình 4 là minh họa trực quan của phương pháp đề xuất.



Hình 4: *Phương pháp đề xuất để nâng cao hiệu quả phân lớp dữ liệu*

Quy trình cân bằng hóa dữ liệu trong phương pháp đề xuất được mô tả như sau. Trước tiên, tập dữ liệu huấn luyện được phân thành các cụm bằng thuật toán K-means, dựa vào độ thưa thớt của mỗi cụm để xác định số lượng mẫu mới cần sinh cho mỗi cụm. Cách làm này để tránh sinh mẫu mới dồn cục vào một khu vực, dẫn đến mất tính đại diện của các mẫu thiểu số. Sau đó, dùng kỹ thuật VCIR, là một mở rộng của CIR để sinh mẫu mới cho mỗi cụm. Với mục đích giảm thiểu ảnh hưởng của nhiễu dữ liệu, trong VCIR chúng tôi đề xuất sử dụng tâm cụm là trọng tâm của mẫu thiểu số thay vì dùng điểm trung bình như của CIR. Việc sinh mẫu mới theo cách này làm cho tập dữ liệu huấn luyện được cân bằng hơn, phân bố đồng đều hơn và các mẫu mới có tính đại diện sát với đặc trưng của dữ liệu thực tế. Việc sinh mẫu mới trong phương pháp đề xuất được hình thức hóa bằng thuật toán Kmeans-VCIR như sau:

Thuật toán sinh mẫu Kmeans-VCIR

Đầu vào: Tập dữ liệu không cân bằng (*DS*) với *m* thuộc tính mô tả bộ dữ liệu X_1, X_2, X_3, …, X_m; r_1, r_2, r_3, …, r_n là các bản ghi

n là số lượng mẫu thiểu số cần tạo

k là số cụm để thực hiện K-Means

irt là ngưỡng cho trước về tỷ lệ không cân bằng giữa 2 lớp

m là số mũ được sử dụng để tính toán mật độ, ở đây được chọn là số các thuộc tính mô tả của mẫu dữ liệu

Đầu ra: Tập dữ liệu cân bằng (*BD*)

Bước 1: Phân cụm tập dữ liệu và lọc các cụm có tỷ lệ mẫu trội và mẫu hiếm theo ngưỡng *irt*

Bước 1.1: Phân cụm K-Means tập dữ liệu với *k* tối ưu (dựa vào hệ số Silhouette)

Bước 1.2: Tính tỉ số cân bằng mỗi cụm theo công thức:

$$TyLeMatCanBang = \frac{SoLuongMauDaSo(c)+1}{SoMauThieuSo(c)+1}$$

Bước 1.3: Lọc các cụm có tỉ số cân bằng < ngưỡng *irt*

Bước 2: Với mỗi cụm *f* được lọc, tính số lượng mẫu thiểu số tăng thêm dựa trên mật độ lớp thiểu số

Bước 2.1: Tính ma trận khoảng cách Euclid của mẫu thiểu số trong mỗi cụm *f*, *MaTranKhoangCach(f).*

Bước 2.2: Tính khoảng cách trung bình của mẫu thiểu số trong mỗi cụm *f*:

KhoangCachTBMauThieuSo(*f*) = *mean*(*MaTranKhoangCach*(*f*))

Bước 2.3: Tính mật độ của mẫu thiểu số: *MatDo*(*f*) = $\frac{Số\ mẫu\ thiểu\ số(f)}{KhoangCachTBMauThieuSo(f)^{m}}$

Bước 2.4: Tính độ thưa thớt của mẫu thiểu số: *DoThuaThot*(*f*) =$\frac{1}{MatDo(f)}$

Bước 2.5: Tính tổng độ thưa thớt của mẫu thiểu số:

TongDoThuaThot = $\sum_f DoThuaThot(f)$

Bước 2.6: Tính toán tỉ lệ sinh của mẫu thiểu số: *TrongSoSinhMau*(*f*) = $\frac{DoThuaThot(f)}{TongDoThuaThot}$

Bước 3: Sinh mẫu mới cho từng cụm *f*

Bước 3.1: Tính toán số lượng mẫu cần sinh cho mỗi cụm *f* theo công thức:

SoLuongMauGiaTang(f) = FLOOR [$n \times$ *TrongSoSinhMau* (*f*)]

Bước 3.2: Chia *f* thành 2 lớp, lớp thiểu số *(f_0)*, lớp đa số *(f_j)*

Bước 3.3: Tính trọng tâm *(C)* của f_0:

$C = \{median(X_1), median(X_2), median(X_3), \dots, median(X_m)\}$

Bước 3.4: Đối với mỗi bản ghi r_i trong f_0 tính khoảng cách Euclid *dist(r_i, C)* từ r_i đến *C*

Bước 3.5: Chọn bản ghi có khoảng cách nhỏ nhất *(f_{min})* đến trọng tâm *C*

Bước 3.6: Tạo *p* số ngẫu nhiên $h_1, h_2, \dots, h_p$ trong khoảng từ 0 đến 1, trong đó *p* = *SoLuongMauGiaTang(f)*

Bước 3.6.1: Tạo bản ghi mới dưới dạng $f_{min} + h_j * C$

Bước 3.6.2: Cập nhật bản ghi mới vào f_0

4. Thử nghiệm và đánh giá kết quả

4.1. Các độ đo hiệu suất phân lớp

Đối với dữ liệu cân bằng, độ chính xác (accuracy) thường được dùng để đánh giá hiệu suất của mô hình phân lớp. Tuy nhiên, đối với tập dữ liệu không cân bằng, độ chính xác thường không phù hợp để đánh giá tính hiệu quả của mô hình. Chính vì vậy, một số độ đo khác sẽ được dùng để đo hiệu năng của mô hình. Các độ đo này sẽ tập trung nhiều hơn vào đánh giá độ chính xác trên nhóm thiểu số. Các độ đo dùng để đánh giá hiệu suất mô hình được tính dựa trên ma trận nhầm lẫn (Bảng 1). Bài báo sử dụng các độ đo F1-Score, G-Mean để đánh giá chất lượng mô hình phân lớp.

Bảng 1: *Ma trận nhầm lẫn*

	Positive dự đoán	**Negative dự đoán**
Positive thực tế	TP	FN
Negative thực tế	FP	TN

Cụ thể, các độ đo sẽ được tính như sau:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Sensitivity = Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

$$Specificity = \frac{TN}{TN + FP} \quad (4)$$

$$G - Mean = \sqrt{Sensitivity * Specificity} \quad (5)$$

Bên cạnh đó, độ đo AUC (Area Under the Curve) cũng được sử dụng để đánh giá chất lượng của mô hình. AUC là chỉ số được tính toán dựa trên đường cong ROC (Receiver Operating Characteristics) nhằm đánh giá khả năng phân loại của mô hình tốt như thế nào. AUC càng gần 1, mô hình phân loại càng tốt.

4.2. Kết quả thử nghiệm

Phương pháp đề xuất được thử nghiệm trên ba bộ dữ liệu không cân bằng từ kho dữ liệu chuẩn quốc tế UCI: Breast_tissue, Vehicle và Ecoli. Các nhãn lớp của các bộ dữ liệu được gom nhóm thành nhóm lớp thiểu số và lớp đa số để thuận lợi hơn khi tính tỉ lệ mất cân bằng đối với bài toán phân lớp nhị phân. Việc gom nhóm này tuỳ vào từng đặc điểm riêng của mỗi bộ dữ liệu. Thông tin về số mẫu, số thuộc tính, số nhãn lớp, tỉ lệ không cân bằng của các bộ dữ liệu được mô tả chi tiết ở Bảng 2.

Bảng 2: *Bộ dữ liệu thực nghiệm từ nguồn dữ liệu chuẩn UCI*

Bộ dữ liệu	Số mẫu	Số thuộc tính	SL nhãn lớp ban đầu	Lớp thiểu số (Nhãn +1)		Lớp đa số (Nhãn -1)		Tỉ lệ MCB (SL đa số/SL thiểu số)
				Số mẫu	Nhãn lớp	Số mẫu	Nhãn lớp	
Breast_tissue	106	9	6	36	fad, car	70	adi, con, gla, mas	1,94
Vehicle	846	18	8	52	pp	284	cp, im, imL, imS, imU, om, omL,	3,25
Ecoli	336	7	3	199	van	647	bus, car	5,46

Áp dụng chỉ số Silhouette để tìm *k* (số cụm) tối ưu cho từng bộ dữ liệu khi sử dụng thuật toán K-Means, thu được bảng số liệu phân cụm với thông số cụ thể được mô tả chi tiết trong Bảng 3.

Bảng 3: *Bộ dữ liệu thực nghiệm từ nguồn dữ liệu chuẩn UCI*

Bộ dữ liệu	Ngưỡng lọc (irt)	Số cụm tối ưu	Cụm	Số mẫu thiểu số	Số mẫu đa số	Tỉ lệ mất cân bằng
Breast_tissue	1,94	2	**1**	**36**	**69**	**1,92**
			2	0	1	

Bộ dữ liệu	Ngưỡng lọc (irt)	Số cụm tối ưu	Cụm	Số mẫu thiểu số	Số mẫu đa số	Tỉ lệ mất cân bằng
Vehicle	3,25	2	**1**	**199**	**370**	**1,86**
			2	0	277	
Ecoli	5,46	3	1	1	104	
			2	3	146	
			3	**48**	**34**	**0,71**

Các cụm có tỉ lệ mất cân bằng bé hơn hoặc bằng ngưỡng lọc *irt* sẽ được giữ lại để thực hiện các bước tiếp theo của thuật toán, các cụm có tỉ lệ mất cân bằng lớn hơn ngưỡng lọc *irt* sẽ bị loại bỏ. Cụ thể hơn, đối với các bộ dữ liệu Breast_tissue, Vehicle và Ecoli, các cụm 1, 1, 3 có tỉ lệ mất cân bằng tương ứng là 1,92; 1,86; 0,71 của từng bộ sẽ được giữ lại, các cụm còn lại sẽ được loại bỏ.

Để đánh giá hiệu quả của phương pháp đề xuất, bài báo thực nghiệm trên các bộ dữ liệu trong Bảng 2 với các phương pháp sinh mẫu thiểu số SMOTE, Borderline-SMOTE (Border-SMOTE), Kmeans-SMOTE, SVM-SMOTE và phương pháp đề xuất Kmeans-VCIR. Thực nghiệm với các mô hình phân lớp phổ biến Decision Tree, KNN và SVM trên ngôn ngữ lập trình Python.

Mỗi bộ dữ liệu được chia ngẫu nhiên theo tỷ lệ 70:30, tức là 70% bản ghi dùng huấn luyện mô hình, 30% còn lại dùng để đánh giá mô hình, kết quả trung bình 10 lần chạy theo các độ đo F1-Score, G-Mean và AUC được trình bày trong Bảng 4.

Bảng 4: *Kết quả thử nghiệm trên các bộ dữ liệu đối với Decision Tree, GaussNB, KNN, SVM*

Bộ dữ liệu	Thuật toán phân lớp	Decision Tree			KNN			SVM		
	Phương pháp tăng mẫu	F1-Score	G-Mean	AUC	F1-Score	G-Mean	AUC	F1-Score	G-Mean	AUC
Breast tissue	SMOTE	0,641	0,720	0,727	0,677	0,752	0,759	0,684	**0,742**	0,758
	Border-SMOTE	0,630	0,712	0,725	**0,720**	**0,786**	**0,789**	0,675	0,736	0,753
	Kmeans-SMOTE	0,616	0,700	0,714	0,683	0,754	0,759	**0,699**	0,737	**0,772**
	SVM-SMOTE	0,629	0,711	0,723	0,675	0,747	0,760	0,654	0,720	0,733
	Kmeans-VCIR	**0,651**	**0,730**	**0,746**	0,644	0,710	0,744	0,625	0,702	0,726
Vehicle	SMOTE	0,857	0,910	0,911	**0,829**	**0,929**	**0,930**	0,921	0,962	0,962
	Border-SMOTE	0,859	0,913	0,914	0,826	0,926	0,928	0,914	0,956	0,957
	Kmeans-SMOTE	0,844	0,898	0,899	0,817	0,908	0,908	**0,922**	0,960	0,960

Bộ dữ liệu	Thuật toán phân lớp	Decision Tree			KNN			SVM		
	Phương pháp tăng mẫu	F1-Score	G-Mean	AUC	F1-Score	G-Mean	AUC	F1-Score	G-Mean	AUC
	SVM-SMOTE	0,843	0,902	0,904	0,828	0,926	0,927	0,919	**0,963**	**0,963**
	Kmeans-VCIR	**0,874**	**0,919**	**0,920**	0,791	0,842	0,850	0,904	0,937	0,938
Ecoli	SMOTE	0,778	0,875	0,880	0,798	0,931	0,932	0,730	0,913	0,914
	Border-SMOTE	0,789	0,883	0,888	0,848	0,940	0,941	0,729	0,889	0,891
	Kmeans-SMOTE	0,784	0,868	0,873	0,857	**0,947**	**0,947**	0,748	**0,919**	**0,920**
	SVM-SMOTE	**0,806**	**0,902**	**0,905**	0,817	0,937	0,938	0,724	0,913	0,914
	Kmeans-VCIR	0,732	0,842	0,849	**0,864**	0,932	0,933	**0,755**	0,858	0,864

Từ kết quả thử nghiệm trong Bảng 4, phương pháp đề xuất cho kết quả tốt hơn những phương pháp khác ở tất cả các độ đo đối với bộ dữ liệu Breast_tissue và Vehicle khi áp dụng giải thuật phân lớp Decision Tree. Cụ thể hơn, kết quả phân lớp theo các độ đo F1-Score, G-Mean, AUC lần lượt là 65,1%, 73%, 74,6% đối với bộ dữ liệu Breast_tissue và 87,4%, 91,9%, 92% đối với bộ dữ liệu Vehicle. Bên cạnh đó, phương pháp đề xuất còn cho kết quả tốt nhất đối với độ đo F1-Score khi dùng các bộ phân lớp SVM và KNN trên bộ dữ liệu Ecoli. Tuy nhiên, đối với các trường hợp khác, thuật toán cho kết quả không tốt hơn.

Thuật toán cây quyết định khá nhạy cảm với nhiễu, kỹ thuật sinh mẫu VCIR dùng giá trị trung vị thay cho giá trị trung bình nên mẫu mới sinh ra có thể sát với đặc điểm của dữ liệu, dẫn đến hiệu suất phân lớp được cải thiện. Tuy nhiên, với Bộ dữ liệu Ecoli phân bố tỷ lệ mất cân bằng giữa các cụm quá lệch nhau nên hiệu suất phân lớp với thuật toán này không cao. Một trong những ưu điểm của các thuật toán SVM và KNN là có khả năng chịu nhiễu tốt nên phù hợp với kỹ thuật sinh mẫu mới SMOTE và các biến thể của nó như Kmeans SMOTE, BorderLine-SMOTE.

Qua kết quả thử nghiệm cho thấy phương pháp đề xuất trong bài báo bước đầu đã có những kết quả khả quan trong việc cải thiện hiệu suất phân lớp trên tập dữ liệu không cân bằng. Đặc biệt là đối với thuật toán phân lớp Decision Tree khi sử dụng bộ dữ liệu Breast_tissue và Vehicle.

5. Kết luận

Bài báo đề xuất một phương pháp để nâng cao hiệu suất phân lớp dữ liệu không cân bằng sử dụng kỹ thuật tăng mẫu thiểu số và đặc trưng mỗi cụm. Kết quả thực nghiệm trên các tập dữ liệu không cân bằng trong lĩnh vực y tế cho thấy, phương pháp đề xuất đạt hiệu suất cao hơn trên một số độ đo so với một số phương pháp phân lớp sử dụng SMOTE và các biến thể của SMOTE. Kỹ thuật sinh mẫu mới sử dụng VCIR và đặc trưng của các

cụm dữ liệu có thể đã tạo ra các mẫu nhân tạo có tính đại diện tốt hơn. Tuy nhiên, một nhược điểm của kỹ thuật này là số mẫu nhân tạo được sinh ra phụ thuộc vào phân bố dữ liệu trong mỗi cụm nên có thể trên một số bộ dữ liệu kết quả phân lớp sẽ không được cải thiện. Một hướng nghiên cứu trong tương lai là tiếp tục thử nghiệm phương pháp sinh mẫu hiếm Kmeans-VCIR với một số thuật toán phân lớp khác như Random Forest, AdaBoost… trên những bộ dữ liệu có tỷ lệ không cân bằng cao hơn.

TÀI LIỆU THAM KHẢO

[1] M. H. A Hamid, M. Yusoff and A. Mohamed, "Survey on Highly Imbalanced Multi-class Data," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, issue 6, 2022. DOI: 10.14569/IJACSA.2022.0130627

[2] Thanh Cong Tran and Tran Khanh Dang, "Machine Learning for Prediction of Imbalanced Data: Credit Fraud Detection," In *15th International Conference on Ubiquitous Information Management and Communication (IMCOM),* 2021. DOI: 10.1109/IMCOM51814.2021.9377352

[3] E.G. Dada et al., "Machine learning for email spam filtering: review, approaches and open research problems," *Heliyon*, vol. 5, issue 6, 2019. DOI: 1s0.1016/j.heliyon.2019.e01802

[4] Fahad Alahmari, "A Comparison of Resampling Techniques for Medical Data Using Machine Learning," *Journal of Information & Knowledge Management (JIKM),* World Scientific Publishing Co. Pte. Ltd., vol. 19, pp. 1-13, 2020. DOI: 10.1142/S021964922040016X

[5] F. Li, X. Zhang, X. Zhang, C. Du, Y. Xu and Y.-C. Tian, "Cost-sensitive and hybrid-attribute measure multidecision tree over imbalanced data sets," *Inf. Sci. (Ny)*, vol. 422, pp. 242-256, 2018. DOI: 10.1016/j.ins.2017.09.013

[6] Q. Cao and S. Wang, "Applying Over-sampling Technique Based on Data Density and Cost-sensitive SVM to Imbalanced Learning," *2011 International Conference on Information Management, Innovation Management and Industrial Engineering*. DOI: 10.1109/ICIII.2011.276

[7] N. V Chawla, K. W. Bowyer, and L. O. Hall, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal Artif. Intell. Res.*, vol. 16, pp. 321-357, 2002. DOI: 10.1613/jair.953

[8] He, H., Bai, Y. and Garcia, E.A, "ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning", In *Proceedings of International Joint Conference on Neural Networks,* Hong Kong, pp. 1322-1328, 2008.

[9] M. Zeng, B. Zou, F. Wei, X. Liu, and L. Wang, "Effective prediction of three common diseases by combining SMOTE with Tomek links technique for imbalanced medical data," *IEEE International Conference of Online Analysis and Computing Science (ICOACS)*, pp. 225-228, 2016. DOI: 10.1109/ICOACS.2016.7563084

[10] H. Han, W. Wang, and B. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning," *Lect. Notes Comput. Sci*., vol. 3644, pp. 878-887, 2005. DOI: 10.1007/11538059_91

[11] H. M. Nguyen, E. W. Cooper, K. Kamei, "Borderline over-sampling for imbalanced data classification," *International Journal of Knowledge Engineering and Soft Data Paradigms*, vol. 3, no. 1, pp.4-21, 2009. DOI: 10.1504/IJKESDP.2011.039875

[12] Douzas G, Bacao F Last F, "Improving Imbalanced Learn-ing Through a Heuristic Oversampling Method Based on K-Means and SMOTE," *Inf. Sci*., vol. 465, pp. 1-20, 2018. DOI: 10.1016/j.ins.2018.06.056

[13] K. K Bejjanki, J Gyani and N Gugulothu, "Class Imbalance Reduction (CIR): A Novel Approach to Software Defect Prediction in the Presence of Class Imbalance", *Symmetry*, vol. 12, issue 3, p. 407, 2020. DOI: 10.3390/sym12030407

[14] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk and F. Herrera, *Learning from Imbalanced Data Sets*, Springer, 2018. DOI: 10.1007/978-3-319-98074-4

[15] Cortes Corinna, Vapnik Vladimir, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273-297, 1995. DOI: 10.1007/BF00994018

ABSTRACT

IMPROVING PERFORMANCE FOR IMBALANCED DATA CLASSIFICATION USING OVERSAMPLING AND CHARACTERISTICS OF EACH CLUSTER

Phan Anh Phong, Le Van Thanh
Vinh University, Nghe An, Vietnam

Received on 19/4/2024, accepted for publication on 21/6/2024

This paper proposes a method to enhance the effectiveness of classifying imbalanced data. The main contribution of the method is integrating the K-means clustering algorithm and the minority oversampling technique VCIR to generate synthetic samples that closely represent the actual data characteristics. Experimental results have shown that the proposed method performs better on several metrics than current popular methods for handling imbalanced data, such as SMOTE, Borderline-SMOTE, Kmeans-SMOTE, and SVM-SMOTE.

Keywords: Data classification; imbalanced data; oversampling; K-Means; SMOTE.